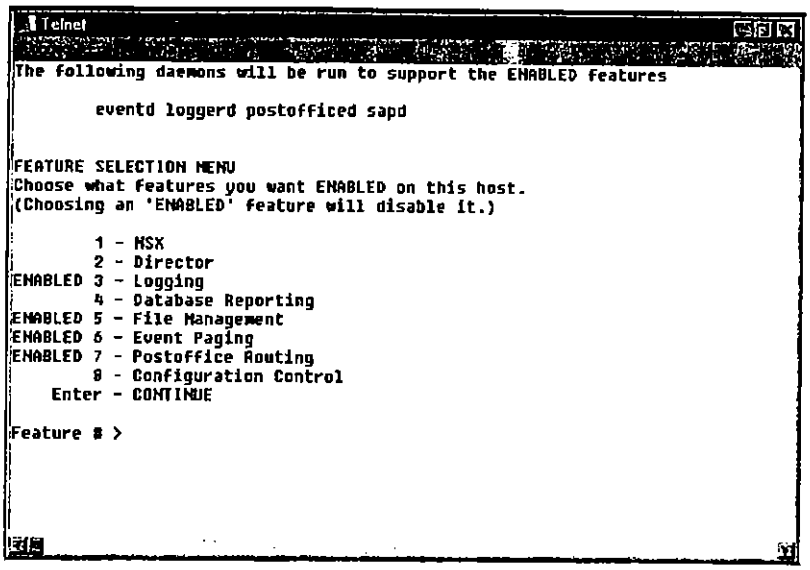
FEATURE SELECTION MENU

This menu, which is shown below, prompts you to select which features you want this installation to support.

When you select a feature from the menu, it is prefixed by "ENABLED" to indicate that it is presently selected. To disable a selected feature, simply choose it from the menu again. The list of daemons required to support the "ENABLED" features is shown at the top of the screen above the Feature Selection menu.



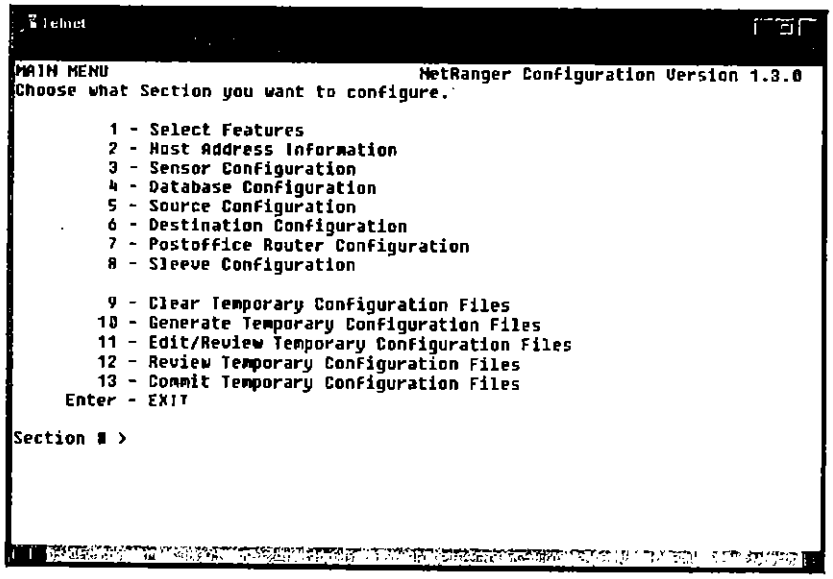
To continue with NetRanger Configuration, choose **Enter** at the menu prompt. This takes you to the Main menu. You can return to this or any other menu at any time from the Main menu.

NOTE

For Standard NSX installation, select features 1 and 5; for standard Director installation, select features 2, 6, and 8.

MAIN MENU

Each menu item takes you through a series of subordinate, or "child", menus that are organized by feature requirements. Features that are not required for the specified NetRanger configuration are preceded by N/A (Not Available). You can return to a configuration menu item at any time by selecting option 1 from the Main menu without losing any information that you may have already input.



The following pages contain worksheets to help you organize your nrconfig information. Each worksheet identifies the required features. You can skip those pages that discuss features that have not been "ENABLED".

1 - Select Features

This menu item takes you to the Feature Selection menu where you can chose which features you want to enable on the NetRanger you are configuring.

2 - Host Address Information **(Required for all Installations)**

LOCAL HOST ADDRESS MENU

Use this menu to enter the Organization names and IDs and Host names and IDs for the NetRanger you are configuring.

Field Name	Description	Your Entry
Organization Name	This is the symbolic name for the Organization ID. The organization name must be identical on all NetRanger systems (NSXs and Directors) within an organization.	
Organization ID	This is the organization ID for the NetRanger you are configuring. You must enter a value between 1-65535 in this field. This ID must be identical on all NetRanger systems (NSXs and Directors) within an organization. To obtain a globally unique organization ID, call WheelGroup Corporation at 888-WGC-NSOC.	
Host Name	A symbolic name for the NetRanger component you are configuring. *This is NOT the network host name.*	
Host ID	A unique ID for the NetRanger you are configuring. *This is NOT the IP address.* You must enter a value between 1-65535 in this field.	

3 - Sensor Configuration **(Required for NSX)**

BORDERGUARD TYPE MENU

Use this menu to enter BorderGuard configuration data in the fields.

Field Name	Description	Your Entry
BorderGuard's Network Host Name	This is the network host name for the BorderGuard used in /etc/hosts on the NSX or in DNS.	
BorderGuard's PASSWORD	This is the password used to log into the BorderGuard.	**DO NOT WRITE YOUR PASSWORD HERE!**
BorderGuard's Version ID/Mode	This is the BorderGuard's Version ID and Configuration Mode. The BorderGuard Version 3 software only supports Router mode. The BorderGuard Version 4 NetSentry software supports either Router or Bridge mode. The default is Version 4 Bridge Mode.	

Based on the **BorderGuard's Version ID/Mode**, the following configuration menus refer either to *Router Mode* or *Bridge Mode Configuration*.

Router Mode Configuration

This section establishes the IP addresses and netmasks for each of the BorderGuard's network interfaces (the BorderGuard should separate your internal protected networks from outside untrusted networks).

The LAN Interfaces Entry menu is the first example of a configuration menu that allows you to list multiple entries. You can add as many entries to the list as you can see on your screen. Each Entry menu allows you to add, edit, or delete entries in the list.

LAN INTERFACES ENTRY MENU

Use this menu to enter BorderGuard configuration information for a LAN interface.

NOTE
It is important to list the outside untrusted network interface first!

Field Name	Description	Your Entry(ies)
Interface's IP Address	This is the IP address used by the BorderGuard on the subnet connected to this interface.	
Netmask	This is the IP mask used on the subnet connected to this interface.	

WAN INTERFACES ENTRY MENU

Use this menu to enter BorderGuard configuration information for a WAN interface.

Field Name	Description	Your Entry(ies)
Interface's IP Address	This is the IP address used by the BorderGuard for this PPP interface.	
PPP IP Address	This is the Destination IP address used by the BorderGuard for this PPP interface.	

BORDERGUARD CONFIGURATION MENU

Use this menu to enter configuration data for a BorderGuard that will be acting as a router.

Field Name	Description	Your Entry
BorderGuard's Primary IP Address*	This is the IP address the NetRanger uses to establish encrypted sleeves for communicating to other NetRanger post offices. This defines the IP address at this end of the encrypted sleeve. Generally this should be the BorderGuard's IP address on the network it is protecting.	
BorderGuard's default gateway	This is the IP address that the BorderGuard should use for its default gateway for IP packet routing.	
Minutes to log on an event	This is the number of minutes to log IP traffic after a serious event occurs. The recommended value is 15 minutes.	
Minutes to shun on an event	This is the length of time in minutes that traffic should be blocked when a serious event occurs. The recommended value is 1440, which equals one day.	
NSX IP Address	This is the IP address used by the NSX on the subnet connecting the NSX to the BorderGuard.	
BorderGuard's IP Address connected to NSX	This is the IP address used by the BorderGuard on the subnet connecting the NSX to the BorderGuard.	
BorderGuard's External IP Address	The NetRanger uses the External IP address as the connection to the untrusted networks.	

NOTE
*If you are using encrypted sleeves over the Internet, this should be a routeable Internet address.

STATIC ROUTES ENTRY MENU

Use this menu to enter the IP addresses, Netmasks, and Gateway IP addresses for the static routes to be implemented by the BorderGuard.

Field Name	Description	Your Entry(ies)
Network's IP Address	This is the IP address for the subnet for the static route.	
Network's Netmask	This is the netmask for the subnet.	
Network's Gateway IP Address	This is the IP address that acts as a gateway to the subnet for the static route.	

Bridge Mode Configuration

BORDERGUARD CONFIGURATION MENU

Use this menu to enter configuration data for a BorderGuard that will be acting as a bridge.

Field Name	Description	Your Entry
BorderGuard's IP Address*	The NetRanger system uses this IP address to establish encrypted sleeves and to communicate with the NSX. This is the BorderGuard's IP address on the subnet connecting the NSX to the BorderGuard.	
BorderGuard's default gateway	This is the IP address that the BorderGuard should use for its default gateway for IP packet routing.	
Minutes to log on an event	This is the number of minutes to log IP traffic after a serious event occurs. The recommended value is 15 minutes.	
Minutes to shun on an event	This is the length of time in minutes that traffic should be blocked when a serious event occurs. The recommended value is 1440, which equals one day.	
NSX IP Address	This is the IP address used by the NSX on the subnet connecting the NSX to the BorderGuard.	

NOTE

*If you are using encrypted sleeves over the Internet, this should be a routeable Internet address.

The following two sensor configuration menus refer to both *Router Mode* and *Bridge Mode* configuration.

SECURITY POLICY CONFIGURATION MENU

Use this menu to establish which incoming services to allow on your interface. You also use this menu to specify the servers to which this traffic will be allowed to pass.

Field Name	Description	Your Entry(ies)
Gateway's IP Address*	This is the IP address of the server that is allowed to service requests coming in through the BorderGuard's External IP address.	
Port	This is the port on the server for the allowed service.	

NOTE

This field supports the definition of multiple IP addresses using any combination of the following formats:

```
ipAddr - a single IP address.
ipAddr,ipAddr - multiple IP addresses.
ipAddr..ipAddr - an inclusive range of IP addresses.
```

Examples: 10.1.6.1, 10.1.6.20, 10.1.6.31, and 10.1.6.35 define 10.1.6.10, 10.1.6.20, and the IP addresses from 10.1.6.31 through 1.01.6.35.

INTERNAL NETWORKS ENTRY MENU

Use this menu to establish the IP addresses and Netmasks for the Internal Protected Networks.

Field Name	Description	Your Entry(ies)
Network's IP Address	This is the IP address for the subnet for the internal network.	
Network's Netmask	This is the netmask for the subnet.	

4 - Database Configuration **(Required for Database)**

DATABASE CONFIGURATION MENU

Use this menu to enter the Database User ID, the Database Password, and the person to be notified for NetRanger events.

Field Name	Description	Your Entry
Database USER ID	This is the user ID used to log into the database.	
Database PASSWORD	This is the password used to log into the database.	**DO NOT WRITE YOUR PASSWORD HERE!**
Notify Person*	This is the person the NetRanger system notifies. Notification is based on criteria you will configure in the sapd.conf file during installation and configuration of NetRanger's *sapd* component. Please refer to *Chapter 5* in this *User's Guide* for additional information.	

NOTE
*This entry must be a valid e-mail or pager address.

5 - Source Configuration **(Required for Director)**

SOURCE ENTRY MENU

Use this menu to enter the Organization and Host names, Organization and Host IDs, and IP routing addresses for the sources of NetRanger events. Enter this information for each NSX that will be sending events to the Director.

Field Name	Description	Your Entry(ies)
Organization Name	This is the source's organization name.	
Organization ID	This is the source's organization ID. This ID must be identical on all NetRanger systems (NSXs and Directors) within an organization. You must enter a value between 1-65535 in this field.	
Host Name	This is the source's host name. This is NOT the network Host name.	
Host ID	This is the source's host ID. You must enter a value between 1-65535 in this field. This is NOT the IP address.	
IP Address to route through	This is the IP address of the NetRanger postoffice that can route NetRanger packets from the source. (If the network does not require an intermediary postoffice, this should be the IP address of the source.)	

6 - Destination Configuration **(Required for NSX/Optional for the Director)**

DESTINATION ENTRY MENU

Use this menu to enter the Organization and Host names, Organization and Host IDs, IP routing addresses, Destination Services, and Event Logging Levels for the destinations of NetRanger events.

Field Name	Description	Your Entry(ies)
Organization Name	This is the destination's organization name.	
Organization ID	This is the destination's organization ID. This ID must be identical on all NetRanger systems (NSXs and Directors) within an organization. You must enter a value between 1-65535 in this field.	
Host Name	This is the destination's host name. *This is NOT the network host name.*	
Host ID	This is the destination's host ID. You must enter a value between 1-65535 in this field. *This is NOT the IP address.*	
IP Address to route through	This is the IP address of the NetRanger postoffice that can route NetRanger packets to the destination. (If the network does not require an intermediary postoffice, this should be the IP address of the destination.)	
Service	This is the name of the service NetRanger events will be sent to. You must enter *loggerd*, *smid*, or *eventd* in this field.	
Level	This is the lowest level of NetRanger alarm/event to send to the service (*loggerd*, *smid*, or *eventd*) you chose in the previous field. You must enter a value between 1-255 in this field. (The recommended level for alarms sent to *smid* is 2 and 1 for alarms sent to *loggerd*.)	

NOTE

On an NSX, you should add the Director machine as a *smid* destination and you should add the NSX machine as a *loggerd* destination so that you can log level 1 alarms/events on the NSX.

7 - Postoffice Router Configuration **(Required for Postoffice routing)**

ROUTER ENTRY MENU

Use this menu to enter the Organization and Host names, Organization and Host IDs, and IP Routing Addresses for remote NetRanger nodes that are not being used as a source or destination.

Field Name	Description	Your Entry
Organization Name	This is the organization name of the remote NetRanger node.	
Organization ID	This is the remote NetRanger node's organization ID. This ID must be identical on all NetRanger systems (NSXs and Directors) within an organization. You must enter a value between 1-65535 in this field.	
Host Name	This is the remote NetRanger node's host name. *This is NOT the network host name.*	
Host ID	This is the remote NetRanger node's host ID. You must enter a value between 1-65535 in this field. *This is NOT the IP address.*	
IP Address to route through	This is the IP address of the NetRanger postoffice that can route NetRanger packets to the remote node. (If the network does not require an intermediary postoffice, this should be the IP address of the remote node.)	

8 - Sleeve Configuration **(Optional for NSX)**

SLEEVED NETWORK ENTRY MENU

Use this menu to enter the Remote Organization ID, Remote IP routing addresses, and Remote Network Netmasks for Sleeved Networks.

Field Name	Description	Your Entry(ies)
Sleeve Remote Org ID	This is the organization ID for the remote end of the sleeve.	
Sleeve Remote IP Address	This is the IP address for the remote end of the sleeve.	
Sleeve Remote Netmask	This is the subnet netmask for the remote end of the sleeve.	

9 - Clear Temporary Configuration Files

This menu item prompts you to ensure that you want to clear the temporary configuration files for the NetRanger software.

```
Are you sure you want to CLEAR the Temporary Configuration files? (y/n)>
```

Choose **y** to clear and reinitialize the temporary NetRanger configuration files in /usr/nr/etc/wgc and the temporary BorderGuard configuration files in /usr/nr/etc/nsc to their default values. This will not discard any configuration information you entered in the current nrconfig session.

10 - Generate Temporary Configuration Files

This menu item prompts you to insure that you want to generate the temporary configuration files for the NetRanger software.

```
Are you sure you want to GENERATE the Temporary Configuration files? (y/n)>
```

Choose **y** to write the temporary NetRanger configuration files containing all the modifications made in the current nrconfig session to /usr/nr/etc/wgc and the temporary BorderGuard configuration files to /usr/nr/etc/nsc.

NOTE

You should review the temporary NetRanger configuration files located in the /usr/nr/etc/wgc directory and the BorderGuard configuration files located in the /usr/nr/etc/nsc directory after nrconfig has generated the temporary configuration data. The temporary NetRanger configuration files **must** be committed to /usr/nr/etc and the temporary BorderGuard configuration files to /tmp after review or after any manual changes. The BorderGuard files must then be loaded onto the NSG BorderGuard via a TFTP session initiated by the BorderGuard.

11 - Edit/Review Temporary Configuration Files

This menu item starts a vi edit on the temporary NetRanger configuration files in /usr/nr/etc/wgc and the temporary BorderGuard configuration files to /usr/nr/etc/nsc.

12 - Review Temporary Configuration Files

This menu item starts a "more" command on the temporary NetRanger configuration files in /usr/nr/etc/wgc and the temporary BorderGuard configuration files in /usr/nr/etc/nsc.

13 - Commit Temporary Configuration Files

This menu item prompts you to ensure that you want to commit the temporary configuration files for the NetRanger software to the NetRanger Configuration File Directory.

```
Are you sure you want to COMMIT the Temporary Configuration files to the
NetRanger Configuration File Directory '/usr/nr/etc' and to the BorderGuard
Configuration File Directory '/tmp'? (y/n)>
```

Choose **y** to write the configuration temporary NetRanger configuration files to the /usr/nr/etc and /tmp directories.

WARNING

This overwrites working NetRanger configuration files.

Enter - EXIT

This menu item prompts you to ensure that you are ready to exit nrconfig. To exit, press **Enter**.

```
Are you sure you want to EXIT? (y/n)>
```

Choose **y** to exit nrconfig.

NOTE

You can exit and restart nrconfig without losing any of the configuration information you have input.

NetRanger network configuration is complete.

NOTE

The NetRanger processes **must** be set using /usr/nr/bin/nrhup before the committed NetRanger configuration files will take effect.

NSX Installations

The BorderGuard Configuration Files **must** be uploaded to the BorderGuard and the BorderGuard **must** be restarted before any changes to the configuration files take effect.

If you are using a BorderGuard 2000 with Version 4.0 of the NetSentry software, then you may not have enough file space on the BorderGuard boot diskette to load the new BorderGuard configuration files. After you make backup copies of the files on the diskette, then you may delete the *readme*, *firewall.def*, and *firewall.set* files which are not needed for normal operation.

Define the Security Policy

Your security policy defines what type of activities and services you want to allow and disallow at key access points on your network. This includes

- services to **allow in** from untrusted sites,
- services to **allow out** from trusted sites (i.e., internal users), and
- services you want to track, such as Web traffic or FTP usage.

Implementation of your security policy is based largely on the filters installed on your BorderGuard or Passport security device. Use the following sections on **ICMP**, **TCP**, and **UDP** protocols as a guide to the configuration of the filter templates. For more information on these services, you may want to refer to such classic texts as *Internetworking With TCP/IP—Volume 1* by Douglas E. Comer (1995).

After defining your security policy, it may be necessary to manually edit the filter files nrconfig places in /tmp, transfer them to the BorderGuard, and then restart the BorderGuard. By default, the incom1.fil filter is applied to the BorderGuard interface on the untrusted network. If you determine that your security policy is identical to the recommended policy outlined in the worksheets on the following pages, then no changes need to be made to the incom1.fil filter file.

WARNING

All manual edits to the files in /usr/nr/etc (the NetRanger configuration files) and /tmp (the BorderGuard configuration files) will be lost if nrconfig is run and the temporary configuration files are committed.

The following information should be available for each configured NetRanger NSX system:

ICMP (Internet Control Message Protocol)

ICMP allows routers and hosts to send error and control messages to other routers or hosts. One of the most frequent uses of this service is in support of the "ping" command, which queries a remote host or network device to see if it is alive on the network. Hackers commonly use this service to discover potential targets by mapping out a remote network. It also allows internal users to check connectivity to a remote site. Use the following chart to help you map allowable ICMP messages.

Enter "A" for Allowed or "B" for Blocked. All incoming requests to your network should be blocked.

Message	Your Entry	Recommendation	Type Field Offset
Echo Reply		Allow	0
Destination Unreachable		Allow	3
Source Quench		Allow	4
Redirect (change a route)		Allow	5
Echo Request		Block	8
Time Exceeded for a Datagram		Allow	11
Parameter Problem on a Datagram		Allow	12
Timestamp Request		Block	13
Timestamp Reply		Allow	14
Address Mask Request		Block	17
Address Mask Reply		Allow	18

NOTE

You may want to unblock services destined for your Internet servers such as a Web server, mail server, or FTP server. This function is explained in greater detail in *Appendix B*.

To adjust the ICMP security policy, edit the INCOM1_ICMP_FAIL filter in tmp/incom1.fil, transfer the edited file to the BorderGuard, and then restart the BorderGuard. (Refer to the *BorderGuard Configuration* section in this chapter for detailed information about transferring files to the BorderGuard from the NSX.)

TCP (Transmission Control Protocol)

TCP is the most common transport layer protocol used on Ethernet and the Internet. Because of the connection-oriented characteristics of TCP, a connection is established every time a TCP service is used. Therefore, it is easy to ***block certain services from entering*** your network while at the same time ***allowing outgoing*** traffic. Use the following chart to help you map allowable TCP services. This list is not all-inclusive, but rather presents the most common TCP services that are included in the filter templates. Services that can be dynamically added to a filter have a corresponding entry in the NSX's *sensord.conf* file.

Enter "A" for Allowed or "B" for Blocked. Also enter the IP Address of the host you want to allow the service to.

Service	Port	Your Entry	Recommendation	IP Address	Filter Name
FTP Reply	Source Port 20*		Allow (All IPs allowed FTP*)		INCOM1_TCP
FTP	21		Allow 1 IP†		INCOM1_21_TCP_FAIL
Telnet	23		Block		INCOM1_23_TCP_FAIL
SMTP (Mail)	25		Allow		INCOM1_25_TCP_FAIL
DNS	53		Allow		INCOM1_53_TCP_FAIL
Gopher	70		Block		INCOM1_70_TCP_FAIL
Finger	79		Block		INCOM1_79_TCP_FAIL
WWW	80		Allow		INCOM1_80_TCP_FAIL
POP2 (Mail)	109		Block		INCOM1_109_TCP_FAIL
POP3 (Mail)	110		Block		INCOM1_110_TCP_FAIL
RPC	111		Block		INCOM1_111_TCP_FAIL
Auth	113†		Allow 1 or more IP‡		INCOM1_113_TCP_FAIL
NNTP (News)	119		Block		INCOM1_119_TCP_FAIL
NTP	123		Block		INCOM1_123_TCP_FAIL
Exec	512		Block		INCOM1_512_TCP_FAIL
Login	513		Block		INCOM1_513_TCP_FAIL
Cmd	514		Block		INCOM1_514_TCP_FAIL
Printer	515		Block		INCOM1_515_TCP_FAIL

Service	Port	Your Entry	Recommendation	IP Address	Filter Name
ntalk	518		Block		INCOM1_518_TCP_FAIL
uucp	540		Block		INCOM1_540_TCP_FAIL
X11	6000–6063‡		Block		INCOM1_X11_FAIL

*TCP Source Port 20 must be allowed to all hosts that need to be able to FTP to the Internet. Most of the time, this includes everyone, so this TCP service is allowed through.

†Restrict these services to only those host(s) that provide them (i.e., Web server, mail server).

‡The authentication service needs to be allowed for connection back to the mail server and occasionally to individual hosts. Many mail services require this service before they will accept mail, so any trusted system sending mail must allow this service back to it. Some Web sites also require this service. Allowing everyone to use this service does not create any known security holes.

To adjust the TCP security policy, edit the appropriate filter in /tmp/incom1.fil, transfer the edited file to the BorderGuard, and restart the BorderGuard. (Refer to the *BorderGuard Configuration* section in this chapter for detailed information about transferring files to the BorderGuard from the NSX.)

UDP (User Datagram Protocol)

Very few UDP services should be allowed between your network and untrusted sites. UDP is a connectionless service, meaning that it is impossible to distinguish between session initiation and general session data. Use the following chart to help you map allowable UDP services. This list is not all-inclusive, but rather lists the most common UDP services that are included in the filter templates. Other services can be added to the filter and will need to have a corresponding entry in the *sensord.conf* file. The configuration of this file is discussed in a later section.

Enter "A" for Allowed or "B" for Blocked. Also enter the IP Address of the host you want to allow the service to.

Service	Port	Your Entry	Recommendation	IP Address	Filter Name
DNS	53		Allow*		INCOM1_UDP
TFTP	69		Block		INCOM1_TFTP_FAIL
RPC	111		Block		INCOM1_UDP_RPC_FAIL
NTP	123		Block		INCOM1_UDP_NTP_FAIL
NetBIOS	137–139		Block		INCOM1_NETBIOS_FAIL
SNMP	161 & 162		Block		INCOM1_SNMP_FAIL
Syslog	514		Block		INCOM1_SYSLOG_FAIL
RIP	520		Block†		INCOM1_UDP_RIP_FAIL
Response Ports	Port # > 1023		Allow‡		INCOM1_UDP

*You will typically want to restrict external access to a single DNS server.

†If using NetRanger between corporate partners or internal business networks, allow RIP or other routing protocols to pass. Work with the network administrator to determine which protocols need to be allowed.

‡The only standard high-level UDP port usually allowed from an untrusted site is to a DNS server. Restrict it to that single IP address and to the UDP source port 53.

To adjust the UDP security polity, edit the appropriate filter in /tmp/incom1.fil, transfer the edited file to the BorderGuard and restart the BorderGuard. (Refer to the *BorderGuard Configuration* section in this chapter for detailed information about transferring files to the BorderGuard from the NSX.)

4 OPERATING NETRANGER

Working with the Director and the NSX

The NetRanger Director is the Graphical User Interface (GUI) for the NetRanger system. The NetRanger Director (also called "the Director")

- provides a graphical, intuitive display of information pertaining to network security violations in real time;
- displays a hierarchical map of the remote NetRanger software and hardware (e.g., the Sensor processes and the NSX hardware) that send security notifications to the Director;
- provides utilities for configuration of the remote NetRanger applications; and
- provides utilities to query the database of historical security events.

The Director uses popular network management platforms like HP OpenView and IBM NetView to display network security information. As a result of this integration, network management personnel do not have to learn multiple-user interface applications and paradigms to perform different network management tasks.

When a process on a remote NSX machine detects a security violation, a notification (called an "event") is sent from the NSX machine to the Director machine. The Director ensures that the machine and application that generated the event are represented on the graphical map, and then, if the event's severity level exceeds a user-definable threshold, the Director creates an Alarm icon on the map. The color of the Alarm icon is based on the severity of the event. The Application and Machine icons also change color, so it is easy to determine at a glance which machine detected the problem. With a few mouse clicks, details about the Alarm (e.g., source and destination IP address) can be displayed. Location functions can be used to locate Alarms with specific properties.

Once an Alarm is diagnosed and addressed, the user can delete the Alarm icon from the user interface. The Application and Machine icons then revert to their previous state.

Architecture

The NetRanger Director is not a single computer program, but is rather a *set* of applications and background processes that work with a network management platform. The diagram below illustrates the data flow between the processes in NetRanger.

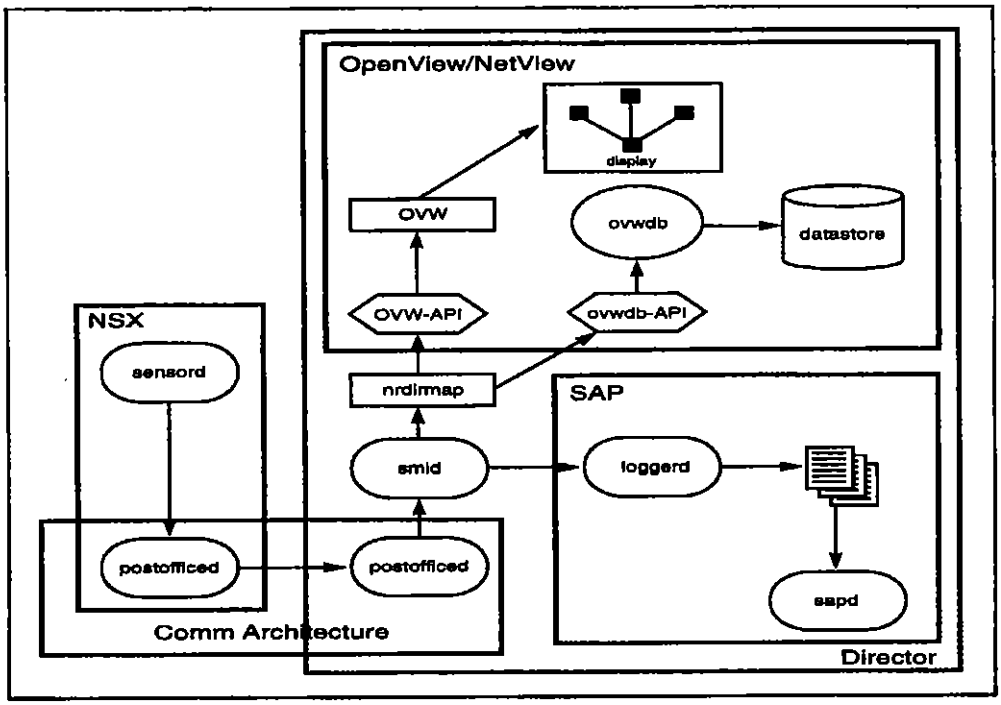


Figure 4.1: The NetRanger Director Architecture

In Figure 4.1, ovals represent background processes, squares represent foreground applications, cylinders represent datastores, hexagons represent APIs, and lines represent the flow of event data. Note that *ovw* and *ovwdb* are part of OpenView/ NetView, *nrdirmap*, *smid*, and *loggerd* are part of the Director, and *sensord* is part of the NSX. Also note that the NSX and the Director both contain *postofficed* processes.

When the *sensord* process detects activity of interest, it generates an event that is sent via the *postofficed* daemons to the *smid* daemon on the Director machine. The *smid* daemon passes the event information to nrdirmap and the *loggerd* daemon, which logs the information.

nrdirmap looks at the severity level of the event. If the event severity exceeds a user-specified level, then *nrdirmap* tells *ovw* to draw an alarm icon. *nrdirmap* also tells *ovwdb* to create an alarm database object in the OpenView/NetView datastore.

Basic Director Functions

Starting the Director

The Director consists of three separate subsystems:

- The NetRanger background processes.
- The network management platform background processes.
- The network management platform user interface.

NOTE

These subsystems should be started in the order listed above to ensure proper operation of the Director.

Starting the NetRanger Background Processes

The NetRanger background processes are configured to start automatically when the machine is rebooted, so in most cases, you will not need to manually start these processes.

In the event that you must start the background processes manually, follow these steps:

1. **Log in as someone in the group** `netrangr`**, and then type**

   `nrstart`

2. **If the executable is not found, then type the fully qualified name,**

   `(/usr/nr/bin/nrstart)`

   put `/usr/nr/bin` **in your path, or type**

   `. /usr/nr/.profile.`

Starting the Network Management Background Processes

Like the NetRanger background processes, the network management platform background processes are configured to start automatically when the machine is rebooted, so in most cases, you will not need to manually start these processes.

In the event that you must start them manually, log in as root and then type `ovstart`

If the executable is not found, then the subdirectory that includes the network management binaries is probably not in your path (the location for OpenView binaries is `$OV_BIN`, and the location for NetView binaries is `/usr/OV/bin`). (To set your path, type `. /usr/nr/.profile`) Consult your network management documentation if you have difficulty starting the network management background processes.

Starting the Network Management User Interface

To start the Director's network management user interface in HP OpenView, log in as a user that belongs to the group `netrangr` and then type

```
ovw &
```

NOTE

The *nrdirmap* program will start automatically when you bring up the network management user interface. You will never have to manually start *nrdirmap*. If you use IBM NetView, type `nv6000 &`

Stopping the Director

To stop the Director, stop the subsystems in the *opposite* order in which they were started.

Stopping the Network Management User Interface

If you use HP OpenView, select **Map...Exit** from the menu.

Usually, you will only want to close the user interface. In most circumstances, you will not want to close the background processes.

If you do want to close the background process, log in as user root and then type `ovstop`

If the executable is not found, then the subdirectory that includes the network management binaries is probably not in your path (the location for OpenView binaries is `$OV_BIN`, and the location for NetView binaries is `/usr/OV/bin`). Consult your network management documentation if you have difficulty starting the network management background processes.

Stopping the NetRanger Background Processes

To stop the NetRanger background processes, follow these steps:

1. **Log in as someone in the group** `netrangr` **and then type**

   ```
   nrstop
   ```

2. **If the executable is not found, then either type the fully qualified name**

   ```
   (/usr/nr/bin/nrstop)
   ```

   or put `/usr/nr/bin` in your path.

Checking the Status of the Director Processes

To check the status of all Director processes, follow these steps:

1. **To ensure that the network management background processes are running correctly, type**

   ```
   ovstatus
   ```

2. **To ensure that the NetRanger background processes are running correctly, type**

   ```
   nrstatus
   ```

   If either of these executables cannot be found, check your path.

Understanding the Director's Submap Hierarchy

When you double-click on a symbol, a submap is opened. This submap could have many symbols on it, and you can double-click these symbols to reveal more submaps. These descending submaps can be thought of as an upside-down tree with more and more branches. This upside-down tree structure is called the "submap hierarchy."

Traversing the submap hierarchy that nrdirmap creates is easy once you understand the following structure:

This type of submap...	can contain these symbols:
Root	Collection
Collection	Machines (NSX and Director)
	Collections
	Connections
Machine	Applications
Application	Alarms
	Alarm Sets

NOTE

Collections and Alarms do not have submaps. They represent the "leaves" in the submap tree.

Figure 4.2 illustrates the root submap. It is the highest level submap in the hierarchy. The root submap has no "parent submap". On the root submap, there should be a symbol representing a Collection of machines.

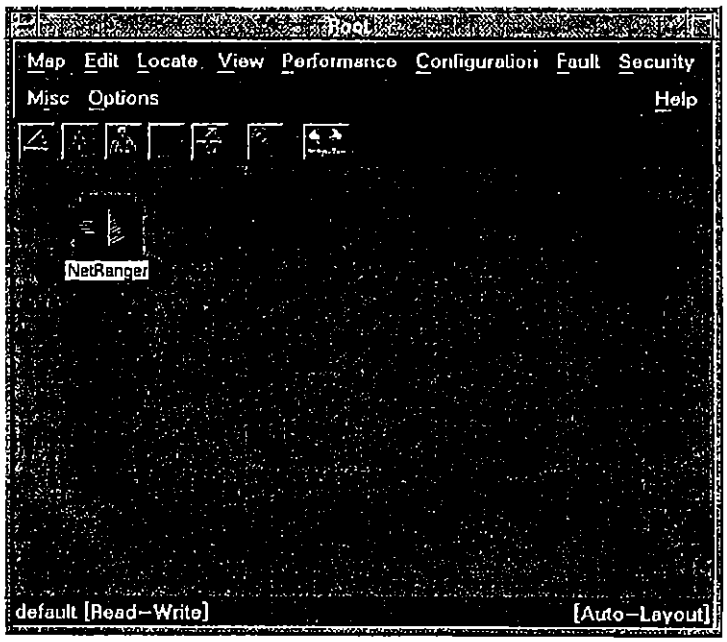


Figure 4.2: The NetRanger Director Submap

When you double-click on a Collection symbol like the one shown in Figure 4.2, a Collection submap is displayed. A Collection submap can have NSX Machines, the Director Machine, other Collection symbols, and Connections between Machines. The Collection submap shown in Figure 4.3 contains a Director machine symbol only

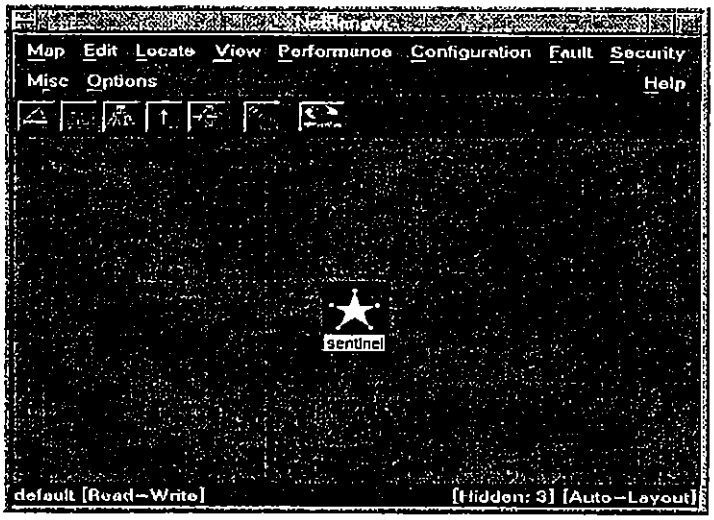


Figure 4.3: A Collection Submap

When you double-click on a Machine symbol, a Machine submap is displayed. A Machine submap contains symbols that represent the different applications running on the machine. Refer to Chapter 1 in this *User's Guide* for an overview of these application daemon services.

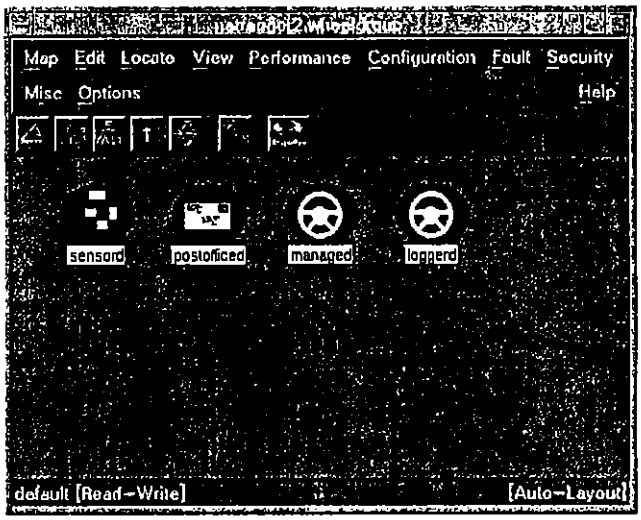


Figure 4.4: A Machine Submap

When you double-click on an Application symbol, an Application submap is displayed. An application submap contains alarms that application had generated. For instance, if the *sensord* application for a machine generates an event and sends it to the Director, the Director will draw an alarm icon on the submap belonging to that machine's application

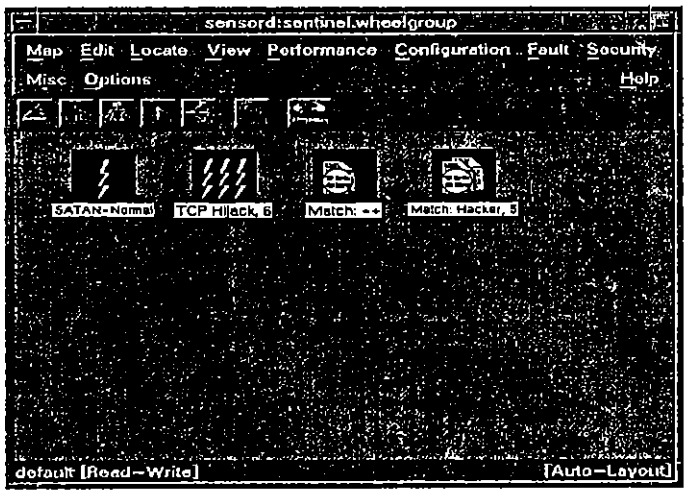


Figure 4.5: An Application Submap

For most alarms, the label under the alarm symbol will match the alarm's "Alarm Name" attribute. For instance, an alarm with the Alarm Name "Net Sweep" will have a "Net Sweep" symbol label.

Alarms with the name "String Match" and "Sec Violation" (Security Violation) will have their symbol labels taken from the "Alarm Details" attribute. This is because there are many types of Security Violations, and there are an infinite number of potential string matches, so for these two alarm types, the Alarm Name itself is not specific enough. For Security Violation alarms, the label will match the name of the specific violation, and for String Matches, the label will be the string that was detected.

Application submaps can also contain a special Alarm symbol called an "Alarm Set". An Alarm Set is created when multiple alarms are received that are identical in all respects *except for* timestamp *and* sequence number. For example, if you get 20 string match alarms with the same attributes (source and destination address, source and destination port etc.), then the 20 alarms will be represented by a single Alarm Set symbol.

Alarm Sets can be differentiated from Alarms in two ways. First, the end of an Alarm Set symbol label will have a comma followed by the number of alarms represented by the Set. Second, the Alarm Set symbol type is slightly different from an Alarm symbol type. An Intrusion Alarm icon has one lightening bolt and an Intrusion Alarm Set has multiple lightening bolts. A String Match Alarm icon has one sheet of paper behind a magnifying glass and a String Match Alarm Set icon has multiple sheets of paper behind a magnifying glass.

If an Application has generated no Alarms, then a special Alarm called an "OkAlarm" (illustrated in Figure 4.6) will be displayed that indicates that the Application has no unresolved Alarms.

Figure 4.6: An OkAlarm

Adding Entities

In general, there are four types of icon symbols: alarms (which include Alarm Sets and OkAlarms), applications, machines, and collections.

Alarm symbols, at the bottom of the submap hierarchy, can only be created by the *nrdirmap* application. An alarm symbol is created whenever an event that exceeds a user-defined threshold is received. There is no way for a user to manually create an alarm symbol.

There are two ways that Application and Machine symbols can be created. First, if an application or host from which an event emanates is not already represented in the map, then *nrdirmap* will create the symbols for you.

If you do not want to wait until an alarm comes in to have a machine or an application represented in a map, you can add the symbols manually. The next two sections describe how to add machines and applications.

Manually Adding an NSX Machine Symbol

To manually add an NSX machine symbol, follow these steps:

1. **Double-click on a Collection symbol to open the Collection submap (the symbol on the root submap labeled "NetRanger" is a Collection symbol). Machines can only be added to Collection submaps! Do not try to add a Machine to a non-Collection submap.**

2. **Select the Edit→Add Object menu function. The Add Object Palette willw appear.**

NetView Users: Select Edit→Add→Object instead.

---

3. **Click on the Net Device icon. Several icons will appear in the bottom of the palette (see Figure 4.7).**

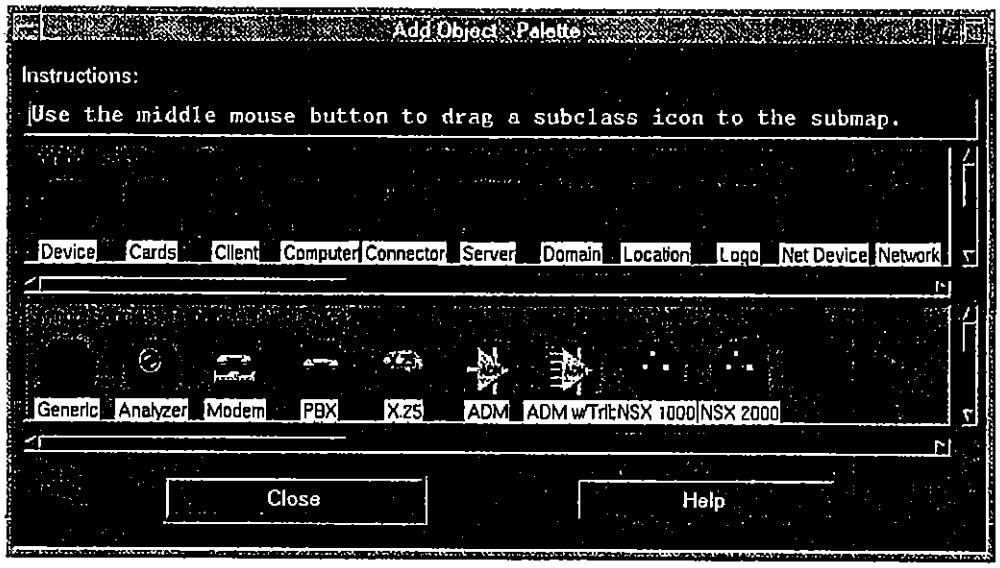


Figure 4.7: The Add Object Palette

4. **Position the mouse pointer over the NSX 2000 icon, press and hold the middle mouse button, and drag the NSX 2000 icon to the collection submap. An Add Object window should appear.**

5. **Select NetRanger/Director from the list, and press the Set Object Attributes button.**

6. **In the hostname field, enter the name of the NSX machine exactly as you entered it in the /usr/nr/etc/hosts file.**

7. **Press the Verify button. If you entered the hostname correctly, NetRanger/ Director will populate the Organization and Host ID fields for you.**

8. **Once the hostname, Organization ID, and Host ID are correct, press OK.**

9. **Press the Set Selection Name button. Choose the selection name from the list and then press OK in the Selection Name window. Press the OK button in the Add Object Window.**

NetView Users: Please skip step #9.

Manually Adding an Application Symbol

1. **Double-click on the NSX Machine to which you wish to add the application. Applications can only be added to Machine submaps! Do not try to add an Application to a non-machine submap!**

2. **If the Add Object Palette is not already displayed, bring it up by selecting the Edit→Add Object menu function.**

NetView Users: Select Edit→Add→Object instead.

---

3. **From the Add Object Palette, click on the WGC Application icon. Several icons should appear in the bottom of the window.**

4. **Using the same technique described above, drag the application icon to the NSX submap.**

5. **Select NetRanger Director from the list, and press the Set Object Attributes button.**

6. **All fields will be populated for you. Press OK.**

7. **Press the Set Selection Name button. Choose the selection name from the list and then press OK in the Selection Name window. Press the OK button in the Add Object Window. You should see the Application icon turn green, because a green OkAlarm will be created automatically in the submap of the Application you just added.**

NetView Users: Please skip step #7.

Manually Adding the Director Machine

Normally, you will not need to manually add the Director Machine. The Director Machine symbol is added for you automatically when nrdirmap comes up the first time. However, it is possible to manually delete the Director Machine, and you may want to manually add the Director Machine back to the map at a later time. Also, if you ever change the Organization ID or Host ID of the Director, then you must delete the Director Machine symbol and add it back with the correct IDs.

NOTE

Only one Director Machine can be represented at a time. In a future release, this restriction may be eased, but for this release, you can only have one Director icon on a map.

The procedure for adding a Director Machine symbol is almost identical to the procedure for adding an NSX Machine symbol. The only differences are as follows:

- From the Object Palette, instead of clicking on the "Net Device" symbol class, click on the "Computer" symbol class.
- Instead of dragging an NSX Machine symbol from the palette, drag the Director Machine icon.
- When you press the Set Object Attributes button, you shouldn't have to enter any data. Unless configuration files (like /usr/nr/etc/hosts) are missing or incorrect on your Director Machine, nrdirmap should be able to fill in this information for you.

Manually Adding an NSX Collection

The **Top-Level NSX Collection** (the entity that appears on the root submap labeled **NetRanger**) is created for you. This is the only Collection that can appear on the root submap. Do *not* try to create additional Collections on the root submap.

NSX Collections are used to customize, or partition, the map. NSX Collections are good tools to use for grouping machines into logical units. See the section in this chapter entitled *How to Customize a Map* for more information about specific uses of NSX Collections.

Follow these steps to add an NSX Collection:

1. **Double-click on the NSX Collection's submap to which you want to add the new NSX Collection. NSX Collections can only be added to NSX Collection submaps! Do not try to add an NSX Collection to a non-NSX Collection submap!**

2. **If the Add Object Palette is not already displayed, bring it up by selecting the Edit→Add Object menu function.**

3. **From the Add Object Palette, click on the Location icon. Several icons will appear in the bottom of the window.**

4. **Using the same technique described above, drag the symbol containing the WheelGroup Logo to the NSX submap.**

5. **Select NetRanger Director from the list, and press the Set Object Attributes button.**

6. **In the NSX Collection Name field, enter the name of the NSX Collection you just moved to the submap. This name can be any unique string. For example, "New York," "Building 162," or "10.1.1 Machines" would be legitimate NSX Collection Names.**

7. **Press the Verify button.**

8. **Press OK.**

9. **Press the Set Selection Name button. Choose the selection name from the list and then press OK in the Selection Name window. Press the OK button in the Add Object Window.**

NetView Users: Please skip step #9.

Modifying Entity Attributes

Once an entity has been created (either by a user or by nrdirmap), it is often helpful to view a listing of the entity's attributes. Some attributes can be edited by the user (for instance, a machine's Point of Contact), and other attributes are read-only (for instance, an alarm's Date). OpenView/NetView provides a visual indication of which fields are changeable and which are not.

To display an entity's attributes in OpenView, select the entity with the mouse pointer, and then select the menu function **Edit→Describe/Modify Object**. You can also select the entity and type Ctrl+O, or you can put the mouse pointer over the icon, press the right mouse button, and select **Describe/Modify Object**. A pop-up window will appear. Select **NetRanger Director** from the list of applications, and then press the **Configure** button.

NetView Users: ***To display an entity's attributes, select the entity with the mouse pointer and then select the menu function Edit→Modify/Describe→Object. You can also select the entity and type Ctrl + O. A pop-up window will appear. Select NetRanger Director from the list of applications and press the Configure button.***

Different entities have different attributes, so each entity will be discussed separately.

NSX Collection Attributes

The NSX Collection Name is the single attribute of an NSX Collection. If you change the Name, and then press **OK** on the appropriate screens, the NSX Collection's symbol label and submap name will change to reflect the new name.

Machine Attributes

Machines have four attributes: **Organization ID**, **Host ID**, **Hostname**, and **Point Of Contact**. The **Hostname** and **Point of Contact** are editable, but the **Org ID** and **Host ID** are not. If you need to change an Org or Host ID, the best thing to do is to delete the machine and then re-add the machine with the correct IDs. If the hostname is changed, the Machine's symbol label will be the part of the hostname up to the first dot (.), and the submap name will be the entire hostname.

If you want to store more information about the Point of Contact than the single field will contain, there are two things you can do. First, you can use the **Object Comments** field to store the additional point of contact information. Second, you could put the point of contact information in a separate trouble-ticketing system.

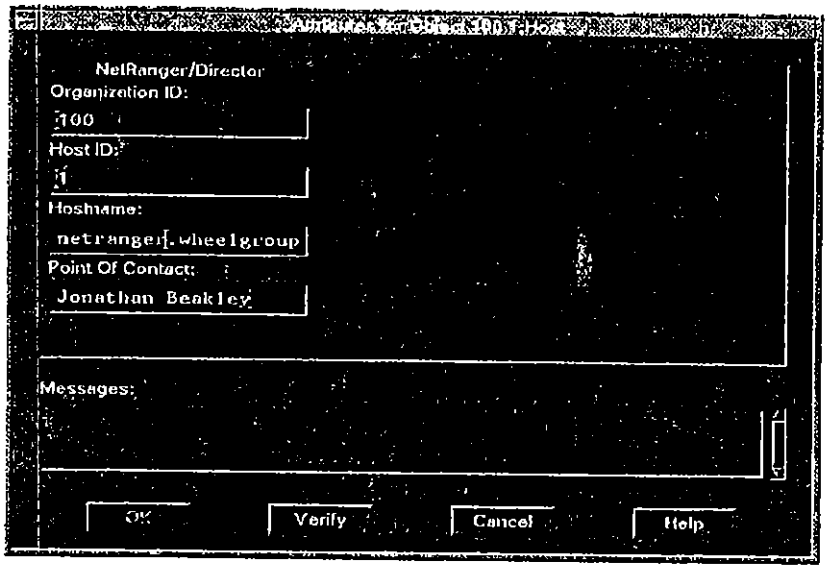


Figure 4.8: Object Attributes Window

Application Attributes

Applications have seven attributes: **Application Name, Minimum Marginal Status Severity, Minimum Critical Status Severity, Alarm Consolidation Threshold, Organization ID, Host ID**, and **Application ID**. The application name, status severity fields, and consolidation threshold are editable, but the ID fields are not. If you need to change an ID, delete the application and then re-add it with the correct IDs. If the application name is changed, the Application's symbol label will change to match the new application name, and the Application's submap name will reflect the new name, too (the format for the Application submap name is **<hostname>:<application name>**).

The **Minimum Marginal Status Severity** describes the lowest severity status an event can have before a marginal (yellow) alarm is created to represent that event. For example, if the minimum marginal status severity is 3, and a severity 2 alarm comes in, then no alarm entity will be created.

NOTE

The higher the severity level, the more severe the alarm. Currently, severity 5 is the highest severity level assigned by the *sensord* daemon.

The **Minimum Critical Status Severity** describes the lowest severity an event can have before a critical (red) alarm is created to represent that event. For example, if the minimum critical status severity is 3, and a severity 4 alarm comes in, then a critical alarm will be created.

NOTE

If you change a status severity value, only events generated **after the change** will be affected. If you increase a threshold severity level from 2 to 3, *nrdirmap* will not remove any existing level 2 alarms from the application's submap. Also, if you decrease a threshold severity level from 3 to 2, nrdirmap will not check historical log files and create alarm icons for severity level 2s that may have occurred in the past. Note that Connections and Alarms do not have submaps.

The **Alarm Consolidation Threshold** describes how many identical alarms must be received before the alarms are replaced by a single "Alarm Set" icon. By default, if two or more alarms are received that are alike in all respects except for timestamp and sequence number, nrdirmap will represent these alarms with a single "Alarm Set" icon. This prevents the screen from being cluttered when many similar alarms are received.

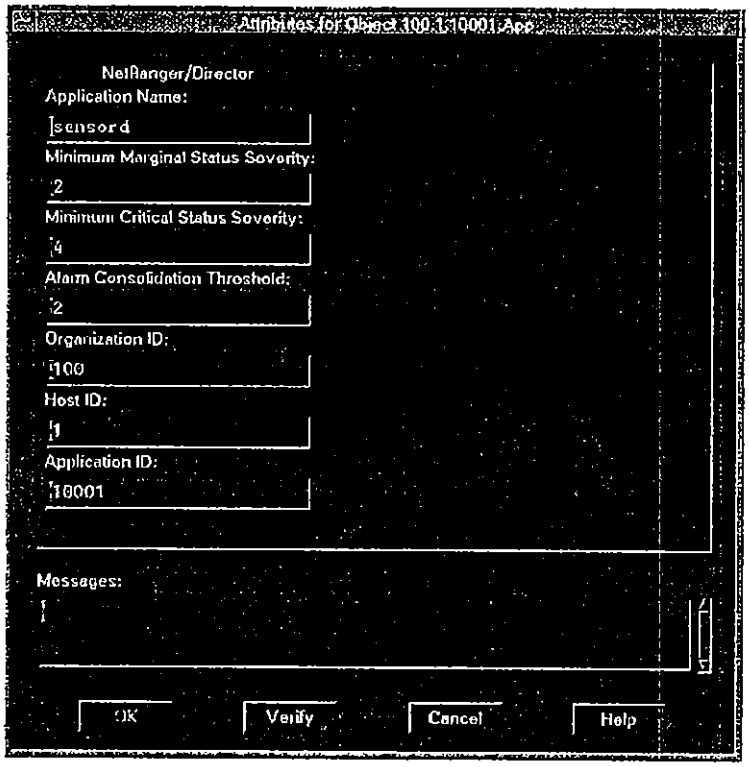


Figure 4.9: sensord Attribute Information

Alarm Attributes

Alarms have many attributes: **Name, Severity, Source Port, Destination Port, Source Address, Destination Address, Router Address, Date, Is Source Address Protected, Is Destination Address Protected, Details, Signature ID, Subsignature ID, Organization ID, Host ID, Application ID, and Instance ID**. All alarm attributes are read-only.

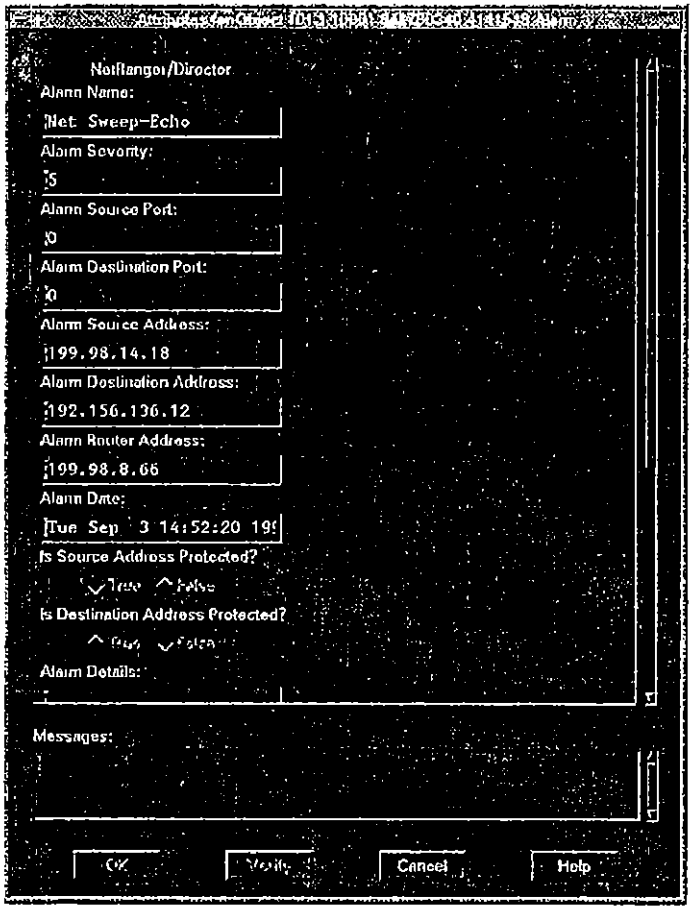


Figure 4.10: Alarm Event Attributes

By default, when two or more alarms are received that are alike in all respects except for timestamp and sequence number, nrdirmap will represent these alarms with a single "Alarm Set" icon. The attributes of an Alarm Set are almost the same as the normal Alarm. The Alarm Set does not have a timestamp and sequence number. Instead, it has an Alarm Count, Date of First Alarm in Set, and Date of Last Alarm in Set.

Once an Alarm Set is created, if additional matching alarms are received, the Alarm Count is incremented, and the Alarm Date(s) are changed if applicable. Note that the symbol label for an Alarm Set is similar to an Alarm, except that after the Alarm Name, the Alarm Count is given.

The special **OkAlarm** that indicates that an application has no unresolved alarms has only four attributes: **Date**, **Organization ID**, **Host ID**, and **Application ID**. All these fields are read-only. The Date field specifies the time at which the **OkAlarm** was created. This gives a lower boundary to the last time that the application in question detected an attack.

Deleting Entities

When you want to remove a symbol (and its corresponding database object), you must select the symbol and then choose the **Edit→Delete→From *All* Submaps** menu option. The most common usage of the delete function is deleting an alarm symbol once the potential hacking attempt has been diagnosed and resolved.

NetView Users: Select Edit→Delete Object→From All Submaps Instead.

There are rules governing the deletion of symbols that help prevent the accidental removal of alarms and other symbols.

NOTE

Applications and Machines can NOT be deleted until ALL of their alarms have been deleted.

This forces the user to go into the submap containing the alarms and specify that it is OK to delete the alarms. This helps prevent a hacking attempt from going unnoticed.

Once an application or host has had all of its alarms resolved (and deleted), you are free to delete the application or machine.

NOTE

If you delete an application or machine, and then an event is received for that machine, the machine will be redrawn on the map. In a case like this, it might be better to unmanage the machine (see the description of the Unmanage function discussed later in this chapter).

Because it would be very easy to accidentally delete large groups of machines, non-empty Collections cannot be deleted. If you have a Collection that contains many machines, and you want to delete the Collection, you must first go into the Collection submap and delete all of the machines (and of course, the machines must have their alarms deleted before the machines can be deleted). Once you have emptied the collection submap, you can then delete the Collection.

NOTE

Never use the Delete Submap function! *nrdirmap* does *not* support this function. Always use the **Delete Object** function to delete entities!

How to Partition a Map

NSX Collection entities can be used to customize, or partition, a map. If the number of NSX machines you are monitoring is too great to represent on a single submap (for instance, the Top-Level NSX Collection submap), you can create additional Collections, and then add Machine icons to those Collection submaps. This allows you to create a hierarchical grouping of machines.

For example, if you had 25 NSX machines in Los Angeles, and 35 machines in New York, you could create an "LA Collection" entity and an "NY NSX Collection" entity. You could then add the NY NSX Machines to the NY Collection, and then add the LA NSX Machines to the LA Collection. This allows you to have fewer symbols per submap, which makes locating symbols and diagnosing problems faster and easier.

NOTE

To put a machine in a collection, you must use the **Add Object** function. If a machine is already represented on the map, and you want to move the representation (the symbol) from one collection to another, you must delete the machine and then re-add it. ***nrdirmap* does *not* support the "Cut and Paste" functionality!** Use of the Cut and Paste functionality on *nrdirmap* entities will yield unexpected results. *You must delete a machine and then re-add it to move the machine symbol from one Collection to another.*

Changing Map Configuration Parameters

There are five global Map-level configuration parameters that can be set. To see these parameters select the menu option **Map→Maps→Describe/Modify**.

NetView Users: ***Select File→Describe Map.***

You will then see a pop-up window. On this window, choose the **NetRanger Director** application, and then press the **Configure** button.

A window with five parameters will appear. You will see the following questions:

1. **Default lowest event severity that generates marginal icon?**
2. **Default lowest event severity that generates critical icon?**
3. **Default Number of Identical Alarms before Icon Consolidation?**
4. **Should nrdirmap be enabled for this map?**
5. **Should new security alarms be shown on the IP Map?**

The answer to the first question specifies the minimum severity an event must have before a marginal (yellow) Alarm is generated. For instance, if you set this value to 2, then if any new applications are created, these applications will have marginal alarms generated for events whose status is two and higher. Of course, if you manually reconfigure the Application symbol to have a new marginal status threshold, then this default value will be overridden.

The answer to the second question specifies the minimum severity an event must have before a critical (red) Alarm is generated. For instance, if you set this value to 3, then if any new applications are created, these applications will have critical alarms generated for events whose status is three and higher. Of course, if you manually reconfigure the Application symbol to have a new critical status threshold, then this default value will be overridden.

The answer to the third question specifies the number of similar alarms that must be received before the alarms are replaced with an "Alarm Set". For instance, if you set this value to 3, then if three identical alarms (identical in all respects except for sequence number and timestamp) are received, then these 3 alarm icons will be replaced with a single Alarm Set whose properties match those of the alarms that were replaced. This function limits screen clutter by reducing the number of icons on the screen.

The answer to the fourth question tells whether nrdirmap is enabled for the currently opened map. By default, all maps will have nrdirmap enabled. If nrdirmap is disabled for a map, then no NetRanger security information will be displayed on that map. This function can be used to control access to NetRanger information.

NOTE

This parameter can only be set when a map is created. Once a map is created, this parameter cannot be changed. For more information on this option, see the section entitled Limiting Access to Security Information.

The fifth question asks if you want alarm icons to be drawn on the IPMap. *This option has no effect with this release. It may be used in a future version.* The IPMap is the submap hierarchy created by the ipmap application. The ipmap application is the part of OpenView/NetView that draws a picture of the IP Topology. The advantage of having alarms represented on the IPMap is that you can view Fault status and Security status on the same screen. The disadvantage is that performance is degraded because extra icons are being created.

NetRanger Director User Interface Menu Functions

On the main menu bar, the menu option called **Security** contains many useful NetRanger Director functions. These functions are described below.

Remotely Configuring NetRanger Daemons

There are two ways to change the configuration of NetRanger daemons running on remote NSX machines.

1. **You can use the nrget/nrset infrastructure to modify daemon characteristics based on "tokens" that are specified by the user. These commands can be run from the command line of the Director machine.**

2. **You can also use the graphical browser utility that comes with the Director. To bring up the utility, select one or more Application or Machine symbols, and then select the menu option Security→Configure.**

   To learn more about nrConfigure, see the nrConfigure section on page 48.

Viewing Alarm Context Information

The **Security→Show→Context** menu function can be used to display data that was being transmitted across your network at the time a security event occurred. Context information is not available for all Alarm types. Currently, only Alarms that involve string matching contain context information.

The following signature IDs correspond to string matching alarms:

Signature ID	Type of string matching
3100-3104	e-mail
3200-3201	WWW
8000	General string matching

All Alarms that have the signature IDs shown above will be represented with "Alarm:Content" icons (icons containing a picture of a magnifying glass over a sheet of paper). These icons differentiate "Content" Alarms from "Intrusion" Alarms (which are represented by icons that contain pictures of lightning bolts).

To use the Show Context function, select one or more Alarm or Alarm Set icon(s), and choose **Security→Show→Context**. An "xnmappmon" window will appear that displays three fields:

- "String Matched". This field displays the string that was matched. The maximum length of this field is 64 bytes.
- "Context Buffer 1". This field displays up to 256 bytes of information that was transmitted in a single direction (either from or to the Server) at the time the string match occurred.
- "Context Buffer 2". This field displays up to 256 bytes of information that was transmitted in the opposite direction (either to or from the Server) at the time the string match occurred.

NOTE

All non-printable ASCII characters are displayed using a "\" character and two hex digits. For example, <ctrl-g>, which has ASCII value "07" in hex, is represented as "\07". The ASCII character "\" itself is represented as "\\".

If there is no context information available for an Alarm, then the "xnmappmon" window will display the following message:

```
Could not find context alarm information for alarm <Alarm Name>.
```

Viewing Event Lists

To view an ASCII list of the latest events that have been generated for a given application or machine, simply select either an Application or a Machine symbol from the map, and then choose the menu option **Security→Show→Current Events**.

This will execute a program that parses the log files in /usr/nr/var, looking for *all* events for the entity selected. Please note that this will include events that may be below the threshold for creating alarms.

Also note that this window is dynamically updated as new events come in. This is why the "hourglass" mouse pointer never goes away. The program does not stop until you press the **Stop** button, because it is always looking for new events.

To stop the search for new events, press the **Stop** button. After you have done this, you can enter new IDs (org/host ID pairs, or org/host/app ID tuples) and restart the search with the **Restart** button. You can also use the various save and print utilities to store the data you have collected.

Press **Stop** and then **Close** to stop the event search and close the window.

The events are displayed with an OpenView/NetView utility called **xnmappmon** (X-node Manager Application Monitor). You can change the fonts and layout of this utility by changing the application defaults file for this utility (see your network management platform documentation for details).

Viewing Database Status and Configuration

The **Security→Show→DB Info** function can be used to show information about the status and configuration of the NetRanger logging and database staging infrastructure.

To use this function, click on a Machine icon that represents a machine running *sapd*, or click on a *sapd* Application icon, and then select **Security→Show→DB Info**. An xnmappmon window will appear with three headings.

Under the heading

```
/usr/nr/var status for <orgId>.<hostId>.<appId>:
```

you will see information about the number and size of files in /usr/nr/var.

Under the heading

```
Run-time history for <orgId>.<hostId>.<appId>:
```

you will see information about the number of database processes that have been run.

Under the heading

```
Trigger configuration for <orgId>.<hostId>.<appId>:
```

you will see database configuration information in the format below.

In the following example, the ORACLE_LOAD trigger is configured such that the DBLoad process will run if the number of DIRFILES in /usr/nr/var/new is greater than or equal to 1.

```
ORACLE_LOAD  DIRFILES  1   /usr/nr/var/new  DBLoad
```

Resolving an Alarm's IP Addresses

The **Security→Show→Names** function can be used to find the hostnames of an alarm's source and destination IP addresses. To use this function, select one or more Alarms (or Alarm Sets), and select **Security→Show→Names**.

An xnmappmon window will display the hostnames if they can be found. If the IP addresses cannot be resolved, you will see the following message:

```
"*** <Resolver> can't find <IP Address>: Non-existent domain"
```

Determining the Version of a Remote NetRanger Daemon

The Director includes a utility called *nrVersion* that determines the Version of NetRanger code running on a machine. This function is helpful when diagnosing problems or upgrading software. To run nrVersion, simply select one or more Machine or Application symbol, and then choose **Security→Show→Version**.

An "xnmappmon" window will display the version numbers of all NetRanger applications that are selected, and/or the version numbers of all NetRanger applications on any machines that are selected.

If you see the following message:

```
Error: Problem sending query to nr.configd
```

then the version of nr.configd that you are using does not support this menu function. You should also ensure that nr.configd is running on the remote machine.

Shunning IP Addresses and Class C IP Networks

The **Security→Shun** functions can be used to manually shun (block) incoming IP traffic. To shun traffic that emanates from an Alarm's Source IP Address, select the Alarm (or Alarm Set), and choose **Security→Shun→Source IP**. To shun traffic that emanates from any IP address within the Class C Network that contains an Alarm's Source IP Address, select the Alarm (or Alarm Set), and select **Security→Shun→Source Net**.

In either case, an xnmappmon window will display the output of the nrexec command that was used to shun the traffic.

Please note that the default timeout value for the nrexec command is 10 seconds, and that the default duration for the shun is 1440 minutes (one day). To extend the duration, to stop the shunning, or to otherwise exercise more granular control, use the **Security→Configure** menu function.

Unshunning IP Addresses and Class C IP Networks

The **Security→Unshun** functions can be used to manually unshun (allow the transmission of) incoming IP traffic. These functions are used to "undo" the effects of the **Security→Shun** functions.

To unshun traffic that emanates from an Alarm's Source IP Address, select the Alarm (or Alarm Set), and choose **Security→Unshun→Source IP**. To unshun traffic that emanates from any IP address within the Class C Network that contains an Alarm's Source IP Address, select the Alarm (or Alarm Set), and select **Security→Unshun→Source Net**.

In either case, an xnmappmon window will display the output of the nrexec command that was used to unshun the traffic.

Please note that if traffic is not already shunned for the selected Alarm, then the Unshun action will have no effect.

Saving Object Data to a File

Use the **Security→Save To File** function to direct the attributes of one or more objects to a file. This is helpful if you want to send an e-mail message about alarm details to someone.

To use this function, select one or more symbols on the map, and then choose **Security→Save To File**. An ASCII file will be created in /usr/nr/tmp. The filename(s) will match the selection name(s) of the object(s) you selected.

This function uses the OpenView/NetView ovobjprint utility. For more information about ovobjprint, see the ovobjprint man page.

Finding Out About nrdirmap

Use the **Security→About** function to find information about the version of nrdirmap that you are using.

Creating a Trouble Ticket

If you have the Remedy ARS (server or client) installed and configured on your Director machine, then you can use the **Security→Trouble Ticket** menu option to create a trouble ticket. If you do not have the Remedy ARS installed, then this menu function will have no effect.

To create a Remedy Trouble ticket, select one or more Alarm or Alarm Set icons and choose the **Security→Trouble Ticket** menu option. One trouble ticket will be created for each icon that is selected. Please note that this means that only one ticket will be created for a selected Alarm Set, even though the Alarm Set represents multiple Alarm notifications. In addition, the date timestamp within the ticket will be the timestamp of the most recent event in the Alarm Set. This function uses NetRanger *sapx* (Security Analysis Package eXtractor) infrastructure to load the data into the Remedy schema. Refer to Chapter 5 in this *User's Guide* to learn how to configure Remedy to work with NetRanger.

Changing IP Addresses, Hostnames, and NetRanger IDs

If you change the IP characteristics of either a Director or NSX machine, or if you change the NetRanger communication infrastructure characteristics (like hostID, orgID, host name, and organization name), you *must* ensure that the appropriate configuration files have been changed on *all* your NetRanger machines, including the Director machine.

If you change the hostID or organizationID of either an NSX machine or the Director machine, after making the necessary changes to the configuration files, ensure that you use the **Edit→Delete** menu option to delete the machine from the map. After this is done, use the **Edit→Add Object** menu function to add the machine back to the map with the proper ids. See the sections on adding and deleting objects for more information about these procedures.

If this changes...	ensure that it is changed here, too (use nrconfig):
an IP Address	/usr/nr/etc/routes
	/usr/nr/etc/sensord.conf
	/usr/nr/etc/managed.conf
host or organization names	/usr/nr/etc/destinations
	/usr/nr/etc/auths
	/usr/nr/etc/hosts
	/usr/nr/etc/routes
	/usr/nr/etc/smid.conf
host or organization IDs	/usr/nr/etc/hosts

If the IP address or hostname of the network management station must be changed, consult your network management documentation to learn about what configuration changes must be made to your network management platform. On HP systems, it is recommended that you shut down the user interface, stop the OpenView daemons, stop the NetRanger daemons, and then use sam to reconfigure the IP information. If you are changing the hostname, you should run `/etc/set_parms hostname` to ensure that the Common Desktop Environment is aware of the new hostname. Once this is done, and once any additional OpenView-specific configuration is complete (as specified in the OpenView documentation), it is recommended that you reboot your machine.

Changing Registration Files

All OVw Applications have a configuration file called a Registration File that tells the User Interface (OVw) how to treat the application. On HP systems, registration files are kept in `$OV_REGISTRATION/C`.

In general, registration files should not be modified, but there are a few circumstances in which it is helpful to edit registration files. Registration files contain the command that is actually used to launch the OVw Application, so registration files are good places to edit an OVw Application's command-line parameters.

The following table lists nrdirmap's command line parameters:

Opt	Params	Function
-a	<integer>	default Alarm consolidation threshold for new maps
-c	<integer>	default Critical value threshold for new maps
-d	<none>	Disable nrdirmap for new maps by default
-f	<none>	force Full synchronization
-k	<none>	Keep symbol copies during delete
-i	<integer>	number of seconds to be Idle before sleep
-m	<non-zero int>	default Marginal value threshold for new maps
-o	<integer>	maximum number of Objects represented in a map
-p	<none>	Propagate to IPMap for new maps
-s	<none>	Secure new map creation
-t	<none>	Tracing enabled

-a, Alarm consolidation threshold

By default, if two or more alarms are received that are alike in all respects except for timestamp and sequence number, nrdirmap will represent these alarms with a single "Alarm Set" icon. This prevents the screen from being cluttered when many similar alarms are received.

An alarm consolidation threshold is configurable for each application object that is represented in the map.

The -a option is used to define a new *default* alarm consolidation threshold. This default value is applied to all application objects that are created *after you change the default value*. For example, if all of the application icons in your map have an alarm consolidation of 2, and then you set the -a option to 5, then whenever a new application is created, the new application will have the threshold of 5. Please note that the -a option will have no effect on *existing* application entities.

The default is 2. A value of 0 means no alarm consolidation. Any integer zero or higher is valid.

-c, Critical value threshold

By default, if nrdirmap receives an event with a severity level of 4 or higher, the symbol that represents that alarm will have "Critical" status. Unless the user specifies otherwise, a symbol with Critical status will be red.

A critical value threshold is configurable for each application object that is represented in the map.

The -c option is used to define a new *default* critical value threshold. This default value is applied to all application objects that are created *after you change the default value*. For example, if all of the application icons in your map have a critical threshold of 4, and then you set the -c option to 3, then whenever a new application is created, the new application will have the threshold of 3. Please note that the -c option will have no effect on *existing* application entities.

-d, Disable nrdirmap by default for new maps

By default, when a new map is created, nrdirmap will be enabled, which means that nrdirmap will display security information on the new map. If you would like for nrdirmap to be disabled for new maps by default, add the "-d" option to the nrdirmap registration file.

See the section in this chapter entitled *Limiting Access to Security Information* for more information about enabling and disabling nrdirmap.

-f, force Full synchronization

By default, when a symbol that exists in multiple maps is deleted from a map, the symbol will not be redrawn on the map when the user interface is stopped and restarted. The idea is that once you delete a symbol from a map, the symbol is deleted permanently. Once a symbol is deleted from *all* maps, the object that the symbol represents is deleted from the object database.

This means that there could be objects in the database that might not be represented as symbols in one or more of your maps (for instance, if you have two maps, and you delete an alarm symbol from one of the maps, the alarm object is still in the database, but there is no symbol for that object in one of your maps).

If you ever want to "refresh" a map to ensure that all objects in the database are represented in a map, use the -f option. This will force nrdirmap to represent all objects in the database as symbols on the map you just brought up.

Note that this option should have no effect if you only have one map.

If you have multiple maps, and you want to "recover" a symbol that was accidentally deleted from a map, you can use this option (assuming that the object is still represented as a symbol in another map).

-l, seconds to be idle before sleep

By default, if nrdirmap does not receive an alarm or a user interface callback for 5 seconds, it will go to "sleep", and wake up once per second to check for new events. Once a new event is received, it will handle new events as fast as they come in, until 5 seconds pass with no more events. At this point, nrdirmap goes back to sleep.

To change the amount of time in seconds before sleep, use this option. Under normal circumstances, there will be no reason to change this value from the default of 5 seconds.

-k, Keep symbol copies during delete

By default, when you delete an entity (for instance, a machine, application, or alarm) from a submap in the Director submap hierarchy, *all* symbols for the entity will be removed from the map. This means that if you make a copy of a symbol and then delete the original symbol, nrdirmap will ensure that the copy is deleted, too. If you want to keep copied symbols on the map when you delete a NetRanger symbol, set the -k option.

WheelGroup recommends the default setting unless you have a specific reason to create and retain symbol copies.

-m, Marginal value threshold

By default, if nrdirmap receives an event with a severity level greater than or equal to 3 and less than the critical threshold value (discussed above), the symbol that represents that alarm will have "Marginal" status. Unless the user specifies otherwise, a symbol with Marginal status will be yellow.

A marginal value threshold is configurable for each application object that is represented in the map.

The -m option is used to define a new *default* marginal value threshold. This default value is applied to all application objects that are created *after you change the default value.* For example, if all of the application icons in your map have a critical threshold of 3, and then you set the -m option to 2, then whenever a new application is created, the new application will have the threshold of 2. Please note that the -m option will have no effect on *existing* application entities.

-o, maximum number of Objects represented in a map

The nrdirmap application creates an object in the OpenView Windows Object Database (ovwdb) for every Alarm (or Alarm Set) icon represented in a map. If the number of objects in the Object Database grows beyond a certain level, OpenView performance can deteriorate and in extreme cases, OpenView can fail.

To prevent these problems, nrdirmap by default will represent no more than 10,000 entities (alarms, machines, applications, etc.) in a map at a given time and no more than 1,000 icons on a given submap (window). If a new event comes in that would cause nrdirmap to exceed either of these threshold values, then nrdirmap will print a warning message to standard out and then redirect the alarm data to a buffer file in /usr/nr/var for safekeeping, thus ensuring the integrity of your OpenView data.

To pull the buffer data from the file into the OpenView user interface, delete enough alarms from the user interface (using the Delete Object menu function) to reduce the number of icons and database objects and then bring the user interface down and back up again. nrdirmap will read the file and create the new alarm icons.

If you are routinely receiving tens of thousands of severe alarms in short periods of time from a given application, you should consider redefining the security policy on your NSX, or increase the Minimum Marginal Status Severity level for that application object.

High-end workstations with plenty of RAM may be able to handle more than 10,000 Objects in a map. If you want to increase (or decrease) this maximum number of Objects value, add the text `-o <value>` to the nrdirmap registration file. It is not recommended that this value exceed 50,000. If you increase this value, you may wish to increase ovwdb's cache size by editing the $OV_LRF/ovwdb.lrf file to improve ovwdb's performance (see the appropriate OpenView documentation for more information).

The 1,000 icons per submap (window) limit is not configurable because this limitation is based on the number of pixels that can be devoted to a window, rather than a value that is user-configurable (such as RAM, disk space, etc.).

-p, Propagate to IP Map

This option currently has no effect. This option may be used in a later version of the product.

-s, Secure map creation

Use this option to ensure that only authorized users can create maps that contain NetRanger information. When the "-s" option is enabled, maps with nrdirmap enabled can only be created from maps that already have nrdirmap enabled. This ensures that a user who does not have access to nrdirmap-enabled maps (and therefore, does not have access to security information) will not be able to create a map that has nrdirmap enabled.

See the section in this chapter entitled Limiting Access to NetRanger Director Security Information on page 44 for more information about enabling and disabling nrdirmap.

NOTE

Enabling the "-s" option will preclude creating nrdirmap-enabled maps from the command line using the `ovw -m <map name>` command. Once the "-s" option is enabled, the only way to create a new map with nrdirmap enabled is to open a map that already has nrdirmap enabled, and then use the **Map→Maps→New** menu function.

NetView Users: Select File→New Map Instead.

-t, Tracing enabled

If nrdirmap is malfunctioning, your authorized service representative might instruct you to enable tracing by adding the -t option. After you set this option, it is best to bring down the user interface, and then bring it back up by typing the following command:

```
ovw > /usr/nr/tmp/nrdirmap.out
```

NOTE

This will create a file called nrdirmap.out with information that can be used by a WheelGroup representative to diagnose the problem. Please note that if you do not redirect the trace messages, the messages will go to standard out.

Command Line Parameter Examples

To enable tracing, type:	`Command -Shared -Initial -Restart "nrdirmap -t";`
To create marginal icons for level 2 alarms and critical icons for level 3 and higher alarms, type:	`Command -Shared -Initial -Restart "nrdirmap -m 2 -c 3";`
To enable tracing and to create Alarm Sets once 15 similar alarms are received, type:	`Command -Shared -Initial -Restart "nrdirmap -t -a 15";`

Changing the Number of Events Displayed in Event List

When you select a Machine or Application symbol and select the menu option **Security→Show Current Events**, by default the last 100 events associated with that entity are displayed (if less than 100 events are known, then all of the events are displayed). To change the number of events that are displayed, use an editor to modify the nrdirmap file, which is stored in `$OV_REGISTRATION/C` on HP and Sun systems, and `/usr/OV/registration/C` on IBM systems.

Replace the "100" with the number of your choice on the line shown in bold.

```
Action show_events  {
        SelectionRule (isWheelGroup && (isMachine || isApplication));
        MinSelected  1;
        Command "sh -c 'unset OVwSessionLoc \;
                $OV_BIN/xnmappmon \
                -selectList \" ${OVwSelections} \" \
                -commandTitle \" Show Current Events for \" \
                -appendSelectList \
                -appendSelectListToTitle \
                -multipleDialogs \
                -headingLine 2 \
                -geometry 900x600 \
                -followOutput \
                -unbufIO \
                -stopSignal 9 \
                -cmd /usr/nr/bin/filterLogByHostApp -l 100' ";
    }
```

Changing Symbol, Object, and Submap Characteristics

The OpenView user interface (ovw) provides functions that modify characteristics of symbols (Icons) and database objects.

Unfortunately, the user interface is perhaps a bit *too* powerful, in that it allows a user to change data in ways that can confuse the nrdlrmap application. This section describes what user modifications are allowed and which are not.

Object Modifications

Object Modifications are made through the **Edit→Describe/Modify Object** menu function.

NetView Users: Select Edit→Modify→Describe→Object.

It is OK to change the Comments field, and it is OK to change any attribute field that is editable from the window that is accessed by selecting NetRanger Director and pressing the **View/Modify Object Attributes** button. Please note that any changes you make to an object *will apply to all maps*. Remember that symbols are map-specific, but database objects are shared among all maps.

You should not edit the Selection Name. The Selection Name field is a field that is used to uniquely identify an object.

Symbol Modifications

Symbol modifications are made through the **Describe→Modify Symbol** menu function.

NetView Users: Select Edit→Modify/Describe→Symbol.

You can change the Display Label setting, and you can change a symbol from explodable to executable.

In general, changing the symbol label itself is not recommended, because nrdirmap has specific algorithms it uses to determine the symbol label, and it will override your label customization when the user interface is stopped and started.

You should not change the symbol status source because nrdirmap expects all symbols (except alarm symbols) to have Compound Status Source. This ensures that an alarm's status is always propagated upward in the submap hierarchy. Changing the status source will jeopardize this.

You should not change the symbol type. This will cause capability fields in the object database to be changed, and this will affect many different functions—including synchronization and callback communication between nrdirmap and ovw. Do not change a symbol's symbol type.

Submap Modifications

Submap modifications are made through the Map→Submap→Describe/Modify menu function.

NetView Users: Select Edit→Modify/Describe→Submap instead.

You can change the Comments and the Background Graphics fields. Changing the Submap Name is not recommended because, like the symbol label, the submap name is set by nrdirmap, and nrdirmap will override your customization when the user interface is shut down and brought back up.

Searching for Symbols

OpenView provides fairly powerful search utilities. These search utilities can be used to locate symbols that match certain criteria. The following three search functions might be useful when searching for alarm symbols:

- Locate by Object Attribute.
- Locate by Symbol Type.
- Locate by Symbol Status.

To use the Locate function, choose **Locate→Objects** from the main menu, and then pick the type of search you want.

For example, to view the number of unresolved **String Matches**, you could search by Symbol Type, and select the **Alarm:Content** symbol type. To determine how many critical elements you have in your network, you could do a search by Symbol Status, and then search for Critical (red) elements. Finally, to search for an alarm from a particular source IP address, you could search by Attribute, and then choose Source IP Address from the list of attributes, and then type in the source IP address you want to find.

Setting the Home Submap

When you start up the user interface, a submap that is designated the home submap is opened. By default, the root (top) level submap is the home submap. You may want to change the home submap to the child submap of the top level NSX Collection (the submap you get when you double-click on the NSX Collection that appears on the root level submap). If you want to make this change, follow these steps:

1. **Double-click the NetRanger Icon.**
2. **Select the menu option Map→Submaps→Set This Submap As Home.**

NetView Users: Select Options→Set Home Submap.

Changing a Submap Background

It is sometimes helpful to place a background picture on a submap to help identify the submap quickly. Submap backgrounds can also be used to help provide context for the different symbols on the submap (for instance, machine icons positioned strategically on a picture of a floor plan could help mark where the machines reside physically).

To add a submap background, open the submap that you want to change, and then select the menu option **Map→Submaps→Describe/Modify**.

NetView Users: Select Edit→Modify/Describe→Submap.

Under the **Background Graphics:** heading, press the **Browse** button.

NetView Users: Press the Select button.

From the pop-up list, select the background graphic of your choice. On HP systems, usastates.gif is a popular choice. You could also create a custom GIF file with any graphics program, and use that GIF file as a submap background. Press **OK** twice.

Repositioning Symbols on a Submap

You can use the mouse pointer to move symbols to different positions on a submap. However, if symbols are added to or removed from the submap, the user interface will automatically reposition *all* of the symbols on the submap, and the customization will be lost.

To prevent this, it is usually best to turn **automatic layout** off. To do this, choose **View→Automatic Layout**, and select the **off** option for either the current submap (if you are only repositioning symbols on a small subset of submaps) or for all submaps (if you reposition symbols frequently).

Hiding Symbols

Under some circumstances, you might want to prevent a symbol from appearing on a given submap, but you might not want, or be able, to delete the symbol. For instance, there could be a machine in a collection that you don't care about, but you can't delete it because it has unresolved alarms. Assume for the moment that there is some reason why you don't want to delete the alarms. In a situation like this, it is best to *hide* the symbols.

To hide a symbol, select it, then choose the menu option **Edit→Hide**.

NetView Users: Select Edit→Hide Objects.

You are given a choice of hiding **This Submap** or **All Submaps**. Pick the option of your choice.

To "unhide" a symbol, simply select the **Edit→Show Hidden Objects** menu function.

Changing the Status Propagation Schemes

When a symbol has Compound Status Source, the status (color) of the symbol is based solely on the status of the symbol(s) in that symbol's child submap. OpenView provides the user with user-selectable sets of "rules" that the User Interface uses to determine the status of a symbol based on the status of the symbols in the child submap. These rulesets are called **Compound Status Source Propagation Schemes**, and the ruleset you choose will affect the color of the icons on the map.

To change the status propagation scheme, choose the **Map→Maps→Describe/Modify** menu option, select one of the radio buttons associated with a scheme, and then press **OK**.

NetView Users: Select File→Describe Map.

WheelGroup Corporation highly recommends that you select the Propagate Most Critical scheme.

Consult the documentation provided with your network management platform for more information about Compound Status Source.

Changing Appearances, Fonts, Window Sizes, Colors, Etc.

In OpenView there are special ASCII files called **Application Default** files that contain parameters that can be customized to change the look and feel of certain applications. To change fonts, window sizes, colors, etc. for the user interface in general, edit the OVw file which is in `$APP_DEFS` on OpenView systems and `/usr/OV/app-defaults` on NetView systems.

To modify the attributes of the various XNm applications, modify the XNm* files. The **Show Current Events** window uses an application called **xnmappmon** (X-Node Manager Application Monitor) to display data. If you want to change the appearance of this window, make the necessary modifications to this file.

Creating and Using Multiple Maps

The NetRanger Director supports the use of multiple OpenView Windows (ovw) maps. Using multiple maps in OpenView can be a little tricky, so consider reading the appropriate OpenView documentation before creating multiple maps.

Here are some things to remember about multiple maps:

- To delete an object from the object database, you must delete the object's symbols from all maps. For example, if an alarm object is represented on two different maps, the alarm must be deleted from both maps before the alarm is cleared from the object database.
- You can create customized maps for different users. Each user's map can have a different subset of NSX Machines displayed. This is helpful when trying to distribute responsibility for different NSX machines to different users.

To create a customized map for a user, either use the **Map→Maps→New** function or use the -m command-line parameter when invoking ovw (see the appropriate OpenView documentation for details).

NetView Users: Select File→New Map.

When the new map is created, nrdirmap will represent all of the objects in the database on the new map. Once the new map has stopped Synchronizing, you can use the **Edit→Delete** function to remove the symbols that you don't want to be represented on that map.

NetView Users: Select Edit→Delete Object.

Usually, Machine symbols are deleted to create a "user domain" with a subset of the configured NSX Machines.

If you delete a Machine from one map, but the Machine is represented in a second map, and if an alarm is generated by that Machine, then the alarm will only be represented on the map containing the Machine.

Please note that if you delete the Machine from *all* maps, then the machine object will be deleted from the database. If that Machine generates a new alarm, then a new database object will be created for that Machine, and the Machine will be represented in *all* maps again.

If you have an NSX Machine that sends alarms to the Director, but you don't want the NSX Machine to be represented in any maps, you should reconfigure the NSX machine to stop sending the alarms to the *smid* process running on the Director. This will prevent nrdirmap from receiving the events and creating the NSX Machine icon. If you still want the events from the NSX Machine to be logged in the Director machine, but not displayed by nrdirmap, configure the NSX Machine to send events to the *loggerd* daemon on the Director rather than the *smid* daemon.

If you want to put back symbols that you previously deleted, there are two things you can do if the objects that the symbols represented are still in the object database.

Option 1: You can use the **Edit→Add Object** menu function to add the object back to the map.

NetView Users: ***Select Edit→Add→Object.***

NOTE

You will get a few warning messages that you are adding an object that already exists. Press **OK** to continue adding the object.

nrdirmap only creates symbols for alarms that are received *after* the machine/application are added to the map. If you want to add a Machine or Application that already exists in the database and you want to also see what alarms already exist, you should use Option 2.

Option 2: If you have many objects that you want to add back to the map, it might be faster to bring the user interface down, and then add the -f option to the nrdirmap registration file. This will force nrdirmap to represent all objects in the database on the map. Refer to the section earlier in this chapter on editing registration files for more information on the -f option.

If one map is open (being viewed with ovw) and a second map is closed, and an event comes in, an alarm will be displayed on the open map. If the alarm is deleted from the open map before the second map is opened, then the alarm will be deleted from the database, and the alarm will not be represented on the second map when the second map is opened. On the other hand, if the second map is opened before the alarm is deleted from the first map, the alarm will be read from the database and will be represented in the second map.

Note that Object attributes apply to all maps. In other words, any time that the **Edit→Describe/Modify Object** option is used to edit an object, these changes will apply to all maps. Unfortunately, there is no way to customize Object attributes on a per-map basis.

NetView Users: Select Edit→Modify/Describe→Object.

Using Read-Only Maps

The NetRanger Director supports the use of read-only user interface sessions. Using read-only maps in OpenView can be a little tricky, so consider reading the appropriate OpenView documentation before using read-only maps.

Here are some things to remember about read-only maps:

- It is not possible to add symbols to or remove symbols from a read-only map. This is a restriction in OpenView that cannot be circumvented. This means that it is impossible for nrdirmap to create an alarm symbol on a read-only map.
- As a result, when an event is received by a read-only map, nrdirmap will change the status of the machine's application's OkAlarm symbol from Normal (green) to the status of the event that was received. This means that even on a read-only map, you should see the status color propagation when an alarm is received.
- Once you see the OkAlarm (and therefore, the Application and Machine) change color on a read-only map, you should use the **Map→Refresh Map** menu option to view an updated copy of the map. This will allow you to view the alarms that have come in.

NetView Users: Select File→Refresh Map.

- Please note that if you have no open read-only copies of the map, then refreshing the map will not provide you with additional information. You must have a read/write copy running to ensure that the map is updated properly.
- If a read-only nrdirmap receives an event, and if the application that generated the alarm does not have an OkAlarm displayed on that read-only map (for instance, if there are other alarms already displayed), then nrdirmap has no OkAlarm symbol to modify in order to reflect the new alarm. As a result, it is advised that users of read-only maps ensure that the maps contain OkAlarms so that status changes are seen immediately. If this is not possible, then read-only users should either refresh the maps occasionally or use *eventd* to notify the user that an event has been received and the map should be refreshed.

Limiting Access to NetRanger Director Security Information

There are functions in the NetRanger Director and in OpenView that can be used together to control which OpenView users have access to security management information.

If you have multiple users who have permissions to run the OpenView user interface, but if you want only a subset of those users to be able to view security information, read the section below.

Limiting access to security information is done by disabling the nrdirmap application for one or more OpenView maps. Once nrdirmap is disabled for a map, nrdirmap will not try to display security information on that map.

Once you have one or more maps that have nrdirmap disabled, and one or more maps that have nrdirmap enabled, you can set the permissions of the OpenView maps to limit which users can access which maps. The result is that only the users with permissions to open the nrdirmap-enabled maps will have the ability to view security information.

Disabling nrdirmap

nrdirmap is disabled on a per-map basis. You must decide whether or not to disable nrdirmap whenever you create a new map.

NOTE

Once a map is created and the decision has been made to enable or disable nrdirmap, the decision is permanent. For instance, once nrdirmap has been disabled for a map, it cannot be enabled.

Disabling nrdirmap can be done in two different ways, depending on how the map is created:

1. **If you create the map from the Map→Maps→New menu option, choose NetRanger Director from the list of configurable applications, and press the Configure button. You will see an option that reads**

   ```
   Should nrdirmap be enabled for this map?
   ```

   Choose `True` to enable nrdirmap and choose `False` to disable nrdirmap. The default is `False` if the -d option appears in the nrdirmap registration file; the default is `True` otherwise.

2. **If you create the map from the command line using the ovw -m option, nrdirmap will be disabled if the -d option appears in the nrdirmap registration file; nrdirmap will be enabled otherwise.**

Setting User Groups

All users who should have access to security information MUST be in the group netrangr. Furthermore, all users who should not have access to security information should NOT be in the group netrangr.

On HP systems, users can be added to and removed from groups using the SAM utility. On Sun systems, the admintool can be used.

NOTE

On HP Systems, if a user is in the group netrangr (because of the configuration of the /etc/group file), but the user's primary group is not netrangr (because the group ID listed in /etc/passwd for the user is not the group ID assigned to the group netrangr), then before executing nrdirmap, the user must type `newgrp - netrangr`. **Sun and IBM users do not have to do this.**

Setting Map Groups and Permissions

Once you have created your maps, you can use the `ovwchgrp` and `ovwchmod` commands to set map permissions.

NOTE

OpenView users should have $OV_BIN in their paths, and IBM users have /usr/OV/bin in their paths.

Maps that have nrdirmap enabled should be readable by users who have access to the /usr/nr subdirectory. This means that maps that have nrdirmap enabled should be owned by the group netrangr, and/or be owned by a user who is in the group netrangr.

Ensure that map permissions are set so that the nrdirmap-enabled maps are not readable by people not in the netrangr group. For example, to allow only the user netrangr and users in the group netrangr to read and write a map called "default", type

```
ovwchown netrangr default
ovwchgrp netrangr default
ovwchmod 660 default
```

Use the `$OV_BIN/ovwls` command to list the maps and verify the permissions.

In the following example, two maps are created. The map called "secinfo" has nrdirmap enabled, and it will be accessible only by the user angie and by other users in the group netrangr. The map called "nosecinfo" has nrdirmap disabled, and it will be accessible only by the user bob and users in the group "staff". Please note that the following example assumes that the -d option has not been added to the nrdirmap registration file.

Set-up

1. **Ensure that user "angie" is in the UNIX group "netrangr".**
2. **Ensure that user "bob" is not in the UNIX group "netrangr".**

Create the "secinfo" map

1. **Log in as user "angie".**
2. **Create the map by typing**

```
ovw -m secinfo &
```

Set the "secinfo" map permissions

1. **From the command line, type**

```
ovwchown angie secinfo
ovwchgrp netrangr secinfo
ovwchmod 660 secinfo
```

Create the "nosecinfo" map

1. **Choose the Map→Maps→New menu option.**
2. **Enter a map name, select "NetRanger/Director" from the list of configurable applications, and press the Configure button.**
3. **Choose `False` in response to the following question:**

```
Should nrdirmap be enabled for this map?
```

4. **Choose OK to create the map.**

Set the "nosecinfo" map permissions

1. **Type the following from the command line:**

```
ovwchown bob nosecinfo
ovwchgrp staff nosecinfo
ovwchmod 660 nosecinfo
```

Now, the map viewable by bob will not contain security information, but the map viewable by angie will.

nrConfigure

Overview

nrConfigure is a Java-based graphical user interface (GUI) that allows you to remotely configure NetRanger applications and access those configurations. It supports all the functionality of the individual nrget, nrgetbulk, nrset, nrunset, and nrexec commands. This GUI-based environment allows you to see an application's tokens, each token's actions, and each action's optional values. Figure 4.11 illustrates the nrConfigure window.

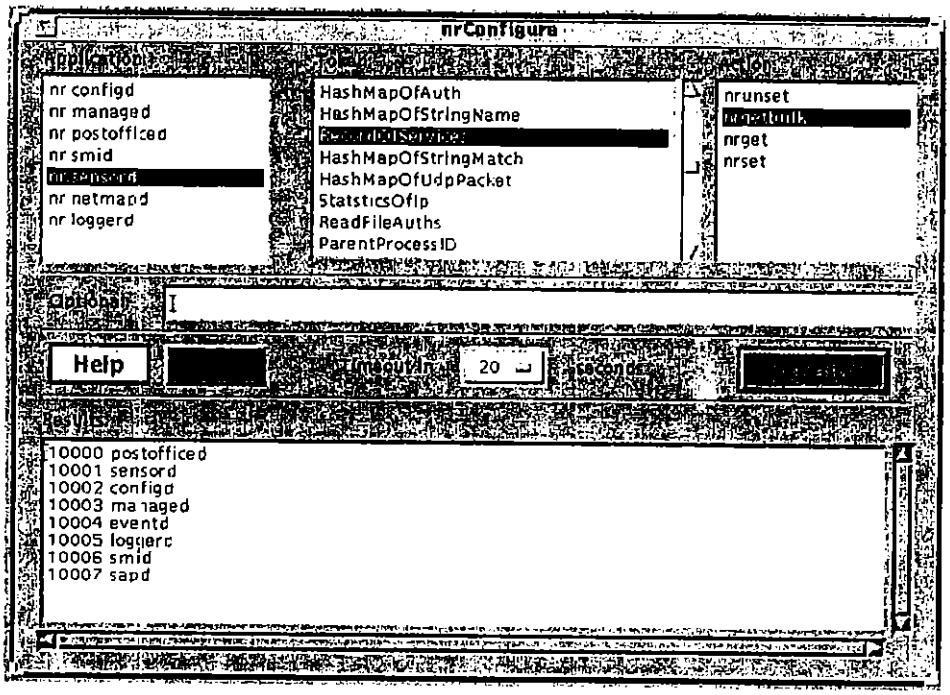


Figure 4.11: The nrConfigure Window

Architecture

The nrConfigure GUI works in the following manner. When you press **Execute** for a given application, token, and action, a packet is sent to *configd*, which sends it on to *postofficed*. *postofficed* sends it on to the proper daemon, which processes the requests and replies. The reply returns through *postofficed* to *configd* and appears in the nrConfigure's Results window.

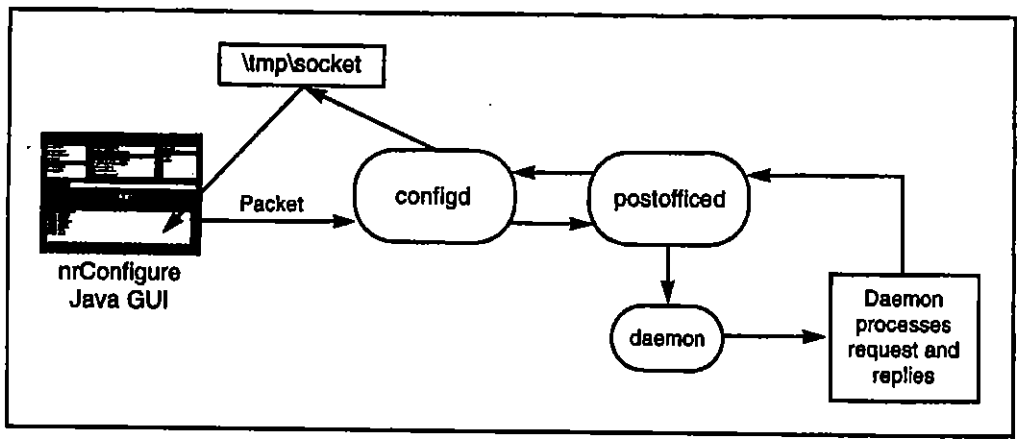


Figure 4.12: nrConfigure Architecture

Starting nrConfigure

nrConfigure is designed to configure daemon applications on an NSX or Director system. This is done by either selecting a machine icon from an OpenView security map or by invoking nrConfigure directly from the command line with the NSX or Director's organization and host Id.

- To start nrConfigure from within the Director, choose **Configure** from the Security menu.
- To start nrConfigure from the NSX command line, type the following command:

```
/usr/nr/bin/nrConfigure <Organization ID>.<Host ID>:<Host Name> &
```

NOTE

The `<Organization ID>` and the `<Host ID>` can be found in /usr/nr/etc/hosts. You may run multiple copies of nrConfigure at the same time.

Quitting nrConfigure

To quit any nrConfigure GUI, press the **Cancel** button. Pressing the **Cancel** button on one GUI will not quit any other nrConfigure GUI session.

Help with nrConfigure

If you need help with nrConfigure, follow these steps:

1. **Choose an Application from the nrConfigure GUI.**

2. **Press the Help Button.**

   A help window is displayed for the selected application. Help is also listed for all of the application-supported tokens.

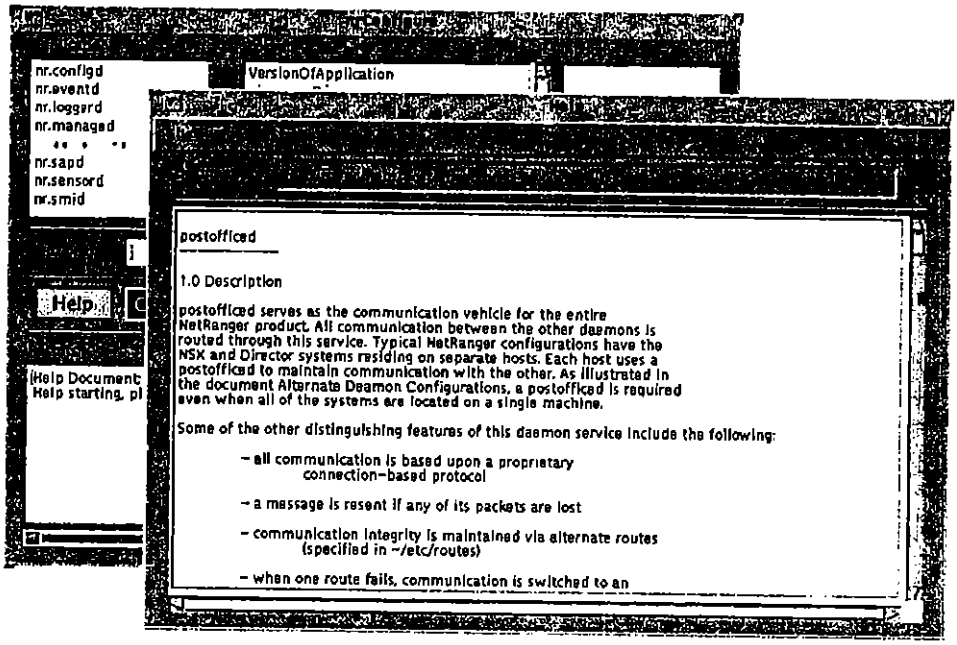


Figure 4.13: nrConfigure Help

Configuring an Application with nrConfigure

To configure an application with nrConfigure, follow these steps:

1. **Select the Application you wish to configure in the Application selection box.**

   The GUI will display that application's allowed Tokens.

2. **Select the Token you wish to execute in the Token Selection box.**

   The GUI will display that token's allowed actions.

3. **Select the Action you wish to execute in the Action selection box.**

The GUI will display the following information:

- Any Optional Parameters, if required
- Any default values for Optional Parameters
- Any set values for Optional Parameters

NOTE

The GUI will display Parameters without current values as the parameter's name in "<>" brackets.

4. **Press the Execute button.**

   The results will display in the Results field.

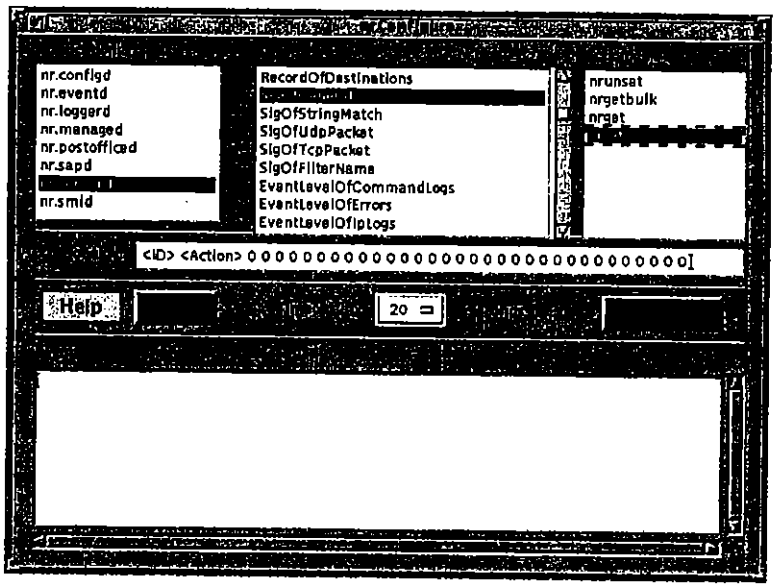


Figure 4.14: Example of Parameters in Optional Field

NetRanger Token Order in the nrConfigure GUI

The display of NetRanger tokens in nrConfigure is dependent upon the order or placement in the list of NetRanger tokens supported by a NetRanger Application or Daemon.

NetRanger Tokens supported by a NetRanger Application are specified in the application persistence file (/usr/nr/etc/persistence.applications). In the following sample, all the NetRanger Tokens are grouped by function (Filename, Read, and Write).

```
APPLICATION=nr.postofficed
ID=10000
HELP=postofficed.txt
TOKEN=FilenameOfError
PARAMETER=filename
DEFAULT=../var/errors.postofficed
TOKEN=FilenameOfConfig
PARAMETER=filename
DEFAULT=../var/postofficed.conf
TOKEN=FilenameOfHosts
TOKEN=FilenameOfServices
TOKEN=FilenameOfAuths
TOKEN=FilenameOfDestinations
TOKEN=ReadFileConfig
TOKEN=ReadFileHosts
TOKEN=ReadFileServices
TOKEN=ReadFileAuths
TOKEN=ReadFileDestinations
TOKEN=WriteFileConfig
END
```

These same NetRanger Tokens could be grouped by the file they operate on (Config, Hosts, Services, Auths, and Destinations), as in the following example:

```
APPLICATION=nr.postofficed
ID=10000
HELP=postofficed.txt
TOKEN=FilenameOfError
PARAMETER=filename
DEFAULT=../var/errors.postofficed
TOKEN=FilenameOfConfig
PARAMETER=filename
DEFAULT=../var/postofficed.conf
TOKEN=ReadFileConfig
TOKEN=WriteFileConfig
TOKEN=FilenameOfHosts
TOKEN=ReadFileHosts
TOKEN=FilenameOfServices
TOKEN=ReadFileServices
TOKEN=FilenameOfAuths
TOKEN=ReadFileAuths
TOKEN=FilenameOfDestinations
TOKEN=ReadFileDestinations
END
```

To edit the location of a NetRanger token in the list of Supported NetRanger tokens for a NetRanger Application, follow these rules:

1. **The APPLICATION, ID, and HELP header (HEADER) must come first to define a valid NetRanger Application.**

```
APPLICATION=nr.postofficed
ID=10000
HELP=postofficed.txt
```

2. **The END tag closes the definition (TRAILER).**

```
END
```

3. **Always move the trailing PARAMETER and DEFAULT tags, if they exist, for the selected token.**

```
TOKEN=FilenameOfError
PARAMETER=filename
DEFAULT=../var/errors.postofficed
```

4. **NetRanger tokens not defined between the HEADER and TRAILER will not show up in the nrConfigure NetRanger token list when the Application is highlighted.**

eventd

Overview

eventd is a general-purpose service that can execute a user-defined action based on alarm events sent to it. These user-defined actions are typically implemented via a shell script, where the various fields of an alarm record are accessible as `$n` command line arguments. As shown in the shell script below, `$1` is the `MessageType`, `$2` is the `RecordID`, `$3` is `GlobalTime`, and so on. Below is an example of the event script in /usr/nr/bin/eventd that parses message fields common to NetRanger message types.

```
# Parse the message fields common to all NetRanger message types
        MessageType=${1}
        RecordID=${2}
        GlobalTime=${3}
        LocalTime=${4}
        DateStr=${5}
        TimeStr=${6}
        ApplID=${7}
        HostID=${8}
        OrgID=${9}
# Parse the message fields based upon the NetRanger message type
        case $(MessageType) in
                2)      ErrorMessage=${10}
                        MessageCount=${11}
                        AlertMessage=`printErrorMessage`
                        ;;
                3)      SrcApplID=${10}
                        SrcHostID=${11}
                        SrcOrgID=${12}
                        CommandMessage=${13}
                        MessageCount=${14}
                        AlertMessage=`printCommandLogMessage`
                           ;;
                4)      SrcDirection=${10}
                        DstDirection=${11}
                        EventLevel=${12}
                        EventSigID=${13}
                        EventSubSigID=${14}
                        ProtocolType=${15}
                        if [ "${ProtocolType}" = "TCP/IP" ]; then
                                SrcIpAddr=${16}
                                DstIpAddr=${17}
                                DstIpPort=${19}
                                SourceAddr=${20}
                                EventMessage=${21}
                                MessageCount=${22}
                        else
                                MessageCount=${16}
                        fi
```

eventd is generally intended to accommodate batch or background processes, such as e-mail notification. Any of the event fields can be used to determine when a user-defined action should occur. For example, a script could be created that generates a separate log of only Level 5 alarms generated by network traffic from a given source IP address.

Using eventd to set up an e-mail notification system

By default, *eventd* is shipped with an e-mail notification service. (Please note that many pager systems can be enabled via e-mail notification.) *eventd* receives copies of alarms from *smid*, arranges them into a readable format, and then sends e-mail messages to users based on information stored in configuration files. An example e-mail message follows:

```
From netrangr@director1.wheelgroup.com Tue Nov  5 12:01:10 1996
From: netrangr@director1.wheelgroup.com
Date: Tue, 5 Nov 1996 12:07:42 -0600
Subject: Alarm OUT->IN Level 2.

Date: 1996/11/05,12:06:00
NSX: [10001.1.100]
Attack: 8000,502 Level: 2
Addrs: 192.216.46.55:80->207.18.164.150:40134
Names: webcrawler.com->lookout.wheelgroup.com
Sig: 8000 "String Match"
Msg: "golf"
```

Basic Setup

eventd is not configured when the package is installed. To set up *eventd*, do the following:

- Set up event script's configuration file
- Set up *eventd* configuration file
- Set up *smid* configuration file
- Set up *eventd* to start

Set Up the Event Script's Configuration File

The event script converts alarms into e-mail messages and sends them to a configurable destination. *eventd* must be configured before it can send alarms. The default configuration file is a template with sample comments only. Unlike the configuration files for NetRanger's daemon services, the configuration files for the event *script* are located in /usr/nr/bin/eventd.

NOTE

The location of the e-mail executables and associated configuration files is totally user-configurable. In this case, these files reside in /usr/nr/bin/eventd. Location and creation of such files is implementation-specific.

Perform the following steps to configure the event script:

1. **Add your organization definition.**

```
ORGANIZATION    100
```

2. **Add users to receive Type 2 alarms. Only one line for Type 2 alarms is accepted.**

```
2       netrangr
```

3. **Add users to receive Type 3 alarms. Only one line for Type 3 alarms is accepted.**

```
3       netrangr
```

4. **Add users to receive Type 4 alarms. Multiple lines may be added for Type 4 alarms. Type 4 alarms have 5 levels, a source, a destination, and recipients.**

```
# Level 4 Alarms
#
# Source and Destination of Alarm
# Src: OUT,IN,-
# Dst: OUT,IN,-
# "-" denotes either
#
#Type  #Level  #Src    #Dst    #Recipients
4      1       OUT     IN      user2
4      2       OUT     IN      user2
4      3       OUT     IN      user2
4      4       OUT     IN      root, user1
4      5       OUT     IN      root, user1
```

- **Level:** The level of the alarm you wish to select. Source and Destination have three possible settings: IN, OUT, and -.
  - **IN.** The address must be inside the network.
  - **OUT.** The address must be from inside the network.
  - **The dash (-) symbol.** The address can be either IN or OUT.
- **Source.** The location of the IP address you wish to select for the recipients

- **Destination.** The location of the IP address you wish to select for the recipients
- **Recipients.** The mail destination of events that match the preceding selection process

Set Up eventd's Configuration File

Before *eventd* can send alarm levels to a destination, the following line in /usr/nr/etc/eventd.conf must be added:

```
EventApplication <UniqueID> <lowestLevelToSend> <locationOfScriptRelativeTo/usr/nr/bin>
```

The following example defines one script for one organization:

```
EventApplication 1002 2 ./eventd/event
```

Set Up smid's Configuration File

In order for *eventd* to receive alarm events from *smid*, the configuration file smid.conf must contain a `DupDestination` entry for *eventd*. An example entry might look as follows:

```
DupDestination director1.wheelgroup eventd  2 ERRORS,COMMANDS,EVENTS
```

Set Up eventd to Start

Edit /usr/nr/etc/daemons by uncommenting the following line:

```
# nr.eventd
```

NOTE

If nr.eventd is left commented or out of /usr/nr/etc/daemons, the daemon will not be started automatically on system start-up.

Advanced Setup

Monitoring Multiple Organizations

If you are monitoring only one (1) organization, use event.conf as your configuration file. Otherwise, build a separate copy of event and event.conf for each organization. To do this, add links pointing to the original event under the name of the organization and make a copy of event.conf under the same name as the new event script. To edit each organization-specific configuration file, follow these steps.

1. **Create a script file with the same name as the configuration file. Use UNIX link command:**

```
%cd /usr/nr/bin/eventd
%ln -s ./event ./event_wheelgroup
```

The script file, when run, will look for a configuration file by the same name with ".conf" appended. Use event.conf as an example.

2. **Make a copy and edit it according to your local needs:**

```
%cp event.conf event_wheelgroup.conf
%vi event_wheelgroup.conf
```

3. **Edit /usr/nr/etc/eventd.conf by adding organization designations, for example:**

```
EventApplication 1002 2 ./eventd/event_wheelgroup
EventApplication 1003 2 ./eventd/event_organization2
```

In the example above, two scripts for two organizations are being added. The organizations are designated as wheelgroup and organization2.

For example, a new script for wheelgroup is made under the name of event_wheelgroup, and event_wheelgroup.conf, respectively.The lines look like this before configuration:

```
-rwxr-x---   1 netrangr netrangr    6603 Nov 22 17:19 event
-rwxr-x---   1 netrangr netrangr    1021 Nov 22 15:30 event.conf
```

and like this after configuration:

```
-rwxr-x---    1 netrangr netrangr      6603   Nov 22 17:19     event
-rwxr-x---    1 netrangr netrangr      1021   Nov 22 15:30     event.conf
lrwxr-xr-x    1 netrangr netrangr      5      Nov 22 15:31     event_wheelgroup-> event
-rwxr-x---    1 netrangr netrangr      1305   Nov 23 12:47     event_wheelgroup.conf
lrwxr-xr-x    1 netrangr netrangr      5      Nov 22 15:31     event_organization2-> event
-rwxr-x---    1 netrangr netrangr      1305   Nov 23 12:47     event_organization2.conf
```

Changing the Event Error Notification user

Event Error Notification is in the event script. It looks for its own configuration file, and if it cannot find it, or it has an error processing the alarm, it notifies the person(s) named as `MAIL_FAILURE`. To change user to receive e-mail on problems with the configuration file, edit `event`, in the /usr/nr/bin/eventd directory, in the following manner:

```
MAIL_FAILURE="userToReceiveEmailOnFailure@somehost.com,netrangr"
```

For example:

```
MAIL_FAILURE="bob@alarmsRus.com,postmaster@walrus"
```

In the above example, "postmaster@walrus" is the second user added to the `MAIL_FAILURE` line.

Event Signatures

Event signatures consist of a unique signature identifier (SigID) and a sub-signature identifier (SubID). This section includes a list of the event signatures currently defined in NetRanger. SigIDs are built into NetRanger; SubIDs change depending upon the SigID they come under.

New Signatures in the 1.3.1 Release

SIGID	Data Context Provided	Level	Description
1100	No	5	IP Fragment Attack—Detects an attack that exploits a vulnerability in TCP/IP stacks with IP fragments
1101	No	2	Unknown IP Protocol—Alarms on any IP packet with a protocol number that RFC1700 declares reserved or unassigned. This range includes 0, 55-60, and 101-255.
2152	No	4	ICMP Flood—Will fire off if the threshold of X ICMP pkts/sec from a single host is exceeded. The SubSigID will be set to the ICMP type code of the last packet.
3105	Yes	3	Sendmail decode—Detects any mail recipient or sender with the name decode. decode is a mail alias that is misused by many intruders to penetrate systems. It does have a legitimate use for automatically decoding mail messages, but may be abused.
3150	Yes	5	FTP Remote command execution—Detects use of the FTP SITE command. This command has been misused to penetrate systems. It does not normally occur within legitimate FTP sessions.
3151	Yes	3	FTP Reconnaissance—Detects use of the FTP SYST command. It provides information about the server that an intruder may use.
3152	Yes	4	FTP CWD ~root exploit—Detects the exploit used to penetrate systems.
3153	Yes	4	FTP PORT command with address other than the client's. Most PORT commands specify a destination address for the server's data connection to be the client. A different address may be specified, but this is rare. Intruders may use this capability for network recon.
3154	Yes	4	FTP PORT command with a reserved port (under 1024). Legitimate data connections do not usually use a reserved port. Intruders may use this capability for network recon.
The next three signatures alert that a URL that ends with one of these three letter extensions is being activated by a user. Users browsing with MS Internet Explorer are vulnerable to these links.			

SIGID	Data Context Provided	Level	Description
3202	Yes	4	User accesses a .url link
3203	Yes	4	User accesses a .lnk link
3204	Yes	4	User accesses a .bat link
3205	Yes	2	HTML file has .url link (context info provided)
3206	Yes	2	HTML file has .lnk link (context info provided)
3207	Yes	2	HTML file has .bat link (context info provided)
3300	No	5	NetBIOS OOB data attack
3500	Yes	5	Attempt to rlogin into system with user name of '-froot'. Detects an exploit used to penetrate AIX systems.
4050	No	3	UDP Bomb—This attack may cause a denial-of-service by crashing the target system the UDP bomb is going to. The SubSigID is the difference in bytes between the IP data length and the UDP datagram length.
6200	Yes	5	ident buffer overflow—Detects improper ident reply from a server that may overflow a buffer, giving access to an intruder.
6201	Yes	5	ident newline—Similar problem to that mentioned above; gives access.
6202	Yes	5	ident improper request—Improper request to an ident server.
The next four signatures indicate that a user has multiple failed authentication attempts.			
6250	Yes	3	FTP authentication failure (3 failed attempts)
6251	Yes	3	Telnet authentication failure (3 failed attempts)
6252	Yes	3	rlogin authentication failure (3 failed attempts)
6253	Yes	3	POP3 authentication failure (3 failed attempts)

Modified Signatures

SigID	Context-based?	Modification
2151	No	Increased size of allowed ICMP traffic
2000–2012	No	SubSigID now contains ICMP code value

TCP/IP Event Signatures

TCP/IP event signatures are currently divided into three groups: **context-**, **content-**, **regular expression-** and **NetSentry-based**.

Context-Based Signatures

Context-based signatures are based on information passed in the TCP/IP header. This can include such things as the destination port, e.g., TCP port 80 for WWW traffic, IP Options, (e.g., source routing), or a combination of events found in a hacking attack. The following group of signatures are context-based attacks that are detectable using NetRanger. Signatures are generated from stateful multi-header packet analysis. NetRanger detects stateful attacks by remembering a sequence of events, or by reconstructing packets to find certain strings or attacks. Those attacks for which alarming is dependent upon maintaining the state of a connection are indicated by a "" following the attack name. These signatures currently analyze the following types of events:

- **Source Routing.** NetRanger detects both loose and strict source routing which is commonly used by hackers to bypass rules found in filtering routers.
- **ICMP Network Sweeps.** This attack uses the ICMP protocol to discover which machines are alive on a remote network. This is most often used as the first step of an attack to find potential targets. This can be implemented three different ways using different ICMP types. All three are detected within NetRanger.
- **Fragmented ICMP traffic.** To get around filtering on large ICMP traffic, it is possible to fragment this traffic and "trick" some intrusion detection systems. NetRanger checks for and alarms on this activity.
- **Large ICMP traffic.** Numerous computers are vulnerable to an attack where if you send an ICMP packet with an extremely large data size it will crash the machine. NetRanger blocks and alarms this traffic.
- **TCP Port Sweep.** When targeting a specific machine, a hacker will frequently run a TCP port sweep to get a list of all available services on the remote target.
- **Half-Open SYN Attack.** This attack was recently publicized when it was used to shut down several Internet Service Providers. This attack can crash a machine by overloading it with TCP connection requests that it never closes.
- **TCP Hijacking.** This attack looks for a characteristic of attacks which take over an existing TCP connection.
- **UDP Port Scan.** When targeting a specific machine, a hacker will frequently run a UDP port sweep to get a list of all available services on the remote target.
- **SATAN Scan.** This looks for both the normal and heavy SATAN attacks.

Content-Based Signatures

For these attacks, NetRanger looks further than simple TCP/IP header information. It actually looks inside the packet for data which indicates an attack in progress. Most of these signatures take advantage of looking for these signatures within a certain context. For example, Sendmail attacks look for certain strings only within the Sendmail port 25.

- **Smail attack.** NetRanger looks for an attack which was only found in the e-mail package "smail".
- **Sendmail invalid recipient.** This looks for attacks which try to send an e-mail to a program on a remote machine. The attack expects the remote machine to execute the e-mail as if it were a program.
- **Sendmail invalid sender.** This attack is like the previous attack except that the sender of the e-mail appears to be a program. When an error occurs within the e-mail the remote machine attempts to return it to the original sender. The e-mail is then executed as a program on the remote machine.
- **Sendmail reconnaissance.** This attack is used to gather information about remote users through the mail port. This is usually used as a preface to other attacks.
- **TFTP password.** This looks for anyone attempting to get the password file using the TFTP service which requires no authentication.
- **DNS HINFO Requests.** This attack uses DNS to gather information about a specific host.
- **DNS Zone Transfer Request.** This attack attempts to gather information about all hosts registered with your DNS server. This can be used by hackers to try to get a map of your network.
- **DNS request for all records.** This attack requests all records maintained on a remote server and it is used to gather information for a future attack.
- **RPC port registration.** This is used to register a specific application to a port on a remote machine and should never occur across the network.
- **RPC port unregistration.** This is used to unregister a specific application to a port on a remote machine and should never occur across the network.
- **RPC dump.** This is used to query a remote machine about which services are running at what ports. This is commonly used by hackers as a preface to an attack.
- **Proxied RPC request.** This is an attack where you trick the remote machine into issuing a request to a service such as NFS for you. This makes the source address appear to be the local machine instead of the attacking machine.

- **NFS *mountd* request.** This is an attack where a remote user tries to connect to the *mountd* process. This attack can determine what file systems a site is sharing on the network and how it might be exploited.
- ***rexd* request.** This is a request to the remote execution daemon and is used to run programs on a remote machine without authentication.
- **YP attacks.** There are five separate attacks which NetRanger looks for which try to take advantage of the service which maintains network passwords and other files.
- **loadmodule attack.** NetRanger looks for a string characteristic of an attack which tricks a set uid program into giving system administrator privileges to the attacker.
- **Any matched string.** NetRanger can look for any string within any service using regular expression matching. This allows any organization to define their own requirements and look for abuse of their proprietary systems.

IP Options Events

IP options consist of a variable-length list of optional information for an IP datagram. Options are rarely used and not all routers and hosts support them. A good security measure is to refuse routing IP datagrams with options. Detecting IP options from externally connected networks provide a good indicator of potential network problems and attacks.

1000 IP options—Bad option list

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

The list of IP options for the datagram is incomplete or malformed. This may indicate an attack where the attacker is using improperly developed hacking software.

1001 IP options—Record packet route

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 2

This option instructs each router to record in the options list the IP address of the interface that the packet will be transmitted from. Because of the limited size of an IP header, only nine addresses can be stored within this list.

1002 IP options—Timestamp

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

This option is similar to the record route option. Routers are requested to add timestamps into the options list. This option is of little value because of the limited size of the options field.

1003 IP options—Provide s, c, h, tcc

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

Basic security options as defined in RFC 1038. Used to carry security level and accrediting authority flags. Implementation of security via these options is obsolete and should not be used.

1004 IP options—Loose source route

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This option specifies a list of IP addresses that the IP datagram must traverse (excluding the routers the datagram can also pass through). An attacker may transmit datagrams into a network using spoofed source addresses that appear to come from the target network. Responses to these packets are transmitted back to the attacker because the host recognizes the source route option. This allows attackers to defeat IP address based authentication mechanisms. Source route attacks usually use *loose routing* due to the number of hops a datagram must traverse across the Internet.

1005 IP options—SATNET ID

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

Stream identifier option. This option is obsolete and should not be encountered.

1006 IP options—Strict source route

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This option specifies the exact list of IP addresses that the IP datagram must traverse. This is similar to loose source routing. Source route attacks usually do not use this type of source routing.

ICMP Events

ICMP datagrams are generated to provide administrative and diagnostic information. The most common use of ICMP datagrams is the "ping" program, which verifies the existence of a host. ICMP traffic can provide much information about a network, and it can also modify network characteristics such as routing tables. Attackers use ICMP to discover and modify this information. SubIDs for the following signatures are set to zero.

2000 ICMP Echo Reply

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

This message is generated in response to an echo request. This type of datagram is sent from the host being pinged.

2001 ICMP Unreachable

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

This message is transmitted to a host stating that the intended destination is unreachable for the IP datagram it transmitted. The first 64 bits of the failed datagram is transmitted along with the unreachable message. Unreachable messages can be used to disrupt existing TCP sessions.

2002 ICMP Source Quench

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 2

This message is transmitted to a host to report network congestion and requests a reduction in the current rate of datagram transmission. While not all systems support this feature, it can be used to enact a denial-of-service attack. The performance of the targeted host can be compromised by sending a continuous stream of these source quench messages.

2003 ICMP Redirect (change a route)

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

This message is transmitted from a router to host with an update for the host's routing table. This enables hosts to start with the minimal routing configuration that is subsequently updated by the network's router tables. Attacker's use redirect messages to place incorrect routes into a target host's routing table to support IP hijacking and other attacks.

2004 ICMP Echo Request

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

This message requests that the destination host transmit back an echo reply message. This type of datagram is sent to the host being pinged. Individual requests are not a security threat. Large numbers of these requests may be a denial of service attack or network reconnaissance as depicted in SigID 2100 below.

2005 ICMP Time Exceeded for a Datagram

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 2

This message was designed to report circular or excessively long routes. A router decrements the time-to-live (TTL) counter whenever it processes a datagram and discards the datagram when the count reaches zero. The first 64 bits of the discarded datagram are transmitted along with the time exceeded message to the originating host. These messages typically occur when an attacker is using the "traceroute" program to help map out a target network. It also occurs when a host has the default TTL set to low (e.g., 30) and transmits datagrams to a destination far across the Internet.

2006 ICMP Parameter Problem on Datagram

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

This message is transmitted when a datagram header is incorrect. The first 64 bits of the incorrect header is also sent.

2007 ICMP Timestamp Request

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 2

This message requests the destination host to transmit back a timestamp reply containing the host's current time. Individual requests are not a security threat. Large numbers of these requests may be a denial of service attack or network reconnaissance as depicted in SigID 2101 below.

2008 ICMP Timestamp Reply

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

The message generated in response to a timestamp request. Use of this feature could be considered as another type of "ping".

2009 ICMP Information Request (obsolete)

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 2

This type of ICMP packet is obsolete and should not be used.

2010 ICMP Information Reply (obsolete)

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

This type of ICMP packet is obsolete and should not be used.

2011 ICMP Address Mask Request

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 2

This message asks the destination host to transmit back the address mask in use on the network. This service was designed to allow diskless clients to set the address mask by broadcasting this request over the local network. Individual requests are not a security threat. Large numbers of these requests may be a denial of service attack or network reconnaissance as depicted in SigID 2102 below.

2012 ICMP Address Mask Reply

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 1

The message generated in response to an address mask request. RFC 950 added this feature, but RFC 1122 forbids a host from sending replies unless it has been explicitly configured as an authoritative agent for address masks. Hosts that do not implement this feature correctly may respond to a targeted request—another type of "ping".

2100 ICMP network sweep w/Echo

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This signature identifies a host that is transmitting ICMP echo request datagrams to multiple hosts on the network. This method is commonly used by attackers to identify active hosts within a network address range. While this can be a serious attack, network management tools such as HP OpenView also perform this type of network discovery on a regular basis.

2101 ICMP network sweep w/Timestamp

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This signature identifies an attacking host that has transmitted ICMP timestamp request datagrams to multiple hosts on the network. While this request can also be used to identify active hosts within a network address range, most attackers use the more common echo request method (see SigID 2100).

2102 ICMP network sweep w/Address Mask

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This signature identifies an attacking host that has transmitted ICMP address mask request datagrams to multiple hosts on the network. While this request can also be used to identify active hosts within a network address range, most attackers use the more common echo request method (see SigID 2100).

2150 Fragmented ICMP Traffic

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 4

This signature identifies an attacking host that has transmitted a fragmented ICMP packet. By design, ICMP packets are small and should never be fragmented. There are attacks that utilize fragmented ICMP packets to crash target systems.

2151 Large ICMP Traffic

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 4

This signature identifies an attacking host that has transmitted a large ICMP packet. By design, ICMP packets are small and should never be fragmented. There are attacks that utilize large ICMP packets to crash target systems.

3000 TCP Connection Logging

SubID Values:0	-	65536	[SYN] Destination TCP port
100000	-	165536	[SYN-ACK] Destination TCP port + 100000
200000	-	265536	[FIN] Destination TCP port + 200000
300000	-	365536	[RST] Destination TCP port + 300000

Misc Field Info: TCP sequence number

Recommended Alarm Value: 1

Packets Required: TCP packets with the SYN, FIN, or RST flags set

This event is used for logging TCP traffic. The token ***LevelOfTrafficLogging*** is used to configure the level of logging.

Level 1: No TCP logging occurs.

Level 2: Only TCP SYN packets are logged (default). This indicates that the source host has initiated an attempt to establish a TCP connection to the destination host using the TCP ports specified. The SubID for this signature is the destination TCP port. If the destination host refuses this connection because the requested port does not exist, an ICMP unreachable message will immediately follow.

Level 3: All TCP SYN, FIN, and RST packets are logged.

The sequence numbers stored in the miscellaneous field provide enough information to determine the number of bytes transferred within a TCP connection. This is accomplished by subtracting the initial sequence number from the SYN packet from the final sequence number from the corresponding FIN packet.

3001 TCP port sweep

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This signature identifies an attacking host that has initiated a series of TCP connections to a number of different destination ports on the target host. This method is used by attackers to determine the services available on the target host for potential exploitation.

3050 Half-open SYN attack

SubID Values: 0

Misc Field Info: Affected TCP port on target system

Recommended Alarm Value: 5

This signature identifies the targeted host and TCP port where the source address and port may be randomly generated by an attacker. Detection of this signature is currently limited to FTP, Telnet, WWW, and e-mail servers.

3100 Smail attack

SubID Values: 0

Misc Field Info: "to: bounce"

Recommended Alarm Value: 4

This signature detects the very common "smail" attack against e-mail servers.

3101 Sendmail Invalid Recipient

SubID Values: 0

Misc Field Info: "to: |"

Recommended Alarm Value: 4

This signature detects any mail message that is transmitted to an address of "pipe" something. Due to the vulnerabilities previously discovered in sendmail and the complexity of the software, destination addresses of this type should not be allowed.

3102 Sendmail Invalid Sender

SubID Values: 0

Misc Field Info: "from: |"

Recommended Alarm Value: 4

This signature detects any mail message that is transmitted with a return address of "pipe" something. This should not be allowed under any circumstance.

3103 Sendmail Reconnaissance

SubID Values: 0

Misc Field Info: "vrfy" or "expn"

Recommended Alarm Value: 2

This represents a reconnaissance attempt by an intruder. It may also represent an advanced user attempting to determine the mailing address of a friend or co-worker.

3104 Archaic Sendmail Attacks

SubID Values: 0

Misc Field Info: "wiz" or "debug"

Recommended Alarm Value: 2

This sendmail attack is archaic and should not work against current versions of Sendmail.

3200 WWW phf attack

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This serious attack is used to exploit the phf program released with the NSCA and Apache web servers.

3201 WWW General cgi-bin attack

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This signature detects any cgi-bin script that attempts to retrieve the file /etc/passwd.

3250 TCP Hijacking

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This signature analyzes both streams of data within a TCP connection to detect when TCP hijacking may have occurred. This current implementation of this signature does not detect all types of TCP hijacking and false positives may occur. Even when hijacking is discovered, little information is available to the operator other than the source and destination addresses and ports of the systems being affected.

4000 UDP packet

SubID Values: 0-65536 Destination UDP port

Misc Field Info: none

Recommended Alarm Value: 1

This event is used for logging UDP traffic. The token ***LevelOfTrafficLogging*** is used to configure the level of logging.

Level 1: No UDP logging occurs.

Level 2: This message records that the source host has sent a UDP datagram to the destination host using the UDP ports specified. The SubID for this signature is the destination UDP port. If the destination host does not have a UDP service at the requested port, an ICMP unreachable message will immediately follow. The default configuration is to not generate this signature unless it is explicitly specified within the configuration file.

Level 3: Same as level 2.

4001 UDP port sweep

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

This could be a serious attack. This signature identifies an attacking host that has sent several UDP datagrams to a number of different destination ports on the target host. This method is used by attackers to determine the services available on the target host for potential exploitation.

4100 TFTP Passwd File

SubID Values: 0

Misc Field Info: none

Recommended Alarm Value: 5

Packets Required: A large sampling of UDP packets

This signature detects an attempt to access the passwd file via TFTP